



การจัดลำดับงานของเครนโดยใช้เทคนิคการโปรแกรมเชิงคณิตศาสตร์และฮิวริสติกส์  
CRANE SCHEDULING BY MATHEMATICAL PROGRAMMING AND  
HEURISTICS

นายชนชัย                      เกตุเทียน                      รหัส 48360502  
นายพรศักดิ์                      วัตสุวิรงค์                      รหัส 48360618

คณะศึกษาศาสตร์
วันที่รับ..... 13 ก.ค. 2553
เลขทะเบียน..... 1506014x
เลขเรียกหนังสือ..... ฟง.....
มหาวิทยาลัยนเรศวร ๖1381

2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมอุตสาหกรรม  
คณะศึกษาศาสตร์ มหาวิทยาลัยนเรศวร  
ปีการศึกษา 2552



## ใบรับรองปริญญาโท

**ชื่อหัวข้อโครงการ** การจัดลำดับงานของเครน โดยใช้เทคนิคการ โปรแกรมเชิงคณิตศาสตร์ และฮิวริสติกส์

**ผู้ดำเนินโครงการ** นายธนชัย เกตุเทียน รหัส 48360502  
นายพรศักดิ์ วัคสูวิงค์ รหัส 48360618

**ที่ปรึกษาโครงการ** ดร. ขวัญนิธิ คำเมือง

**สาขาวิชา** วิศวกรรมอุตสาหการ

**ภาควิชา** วิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

**ปีการศึกษา** 2552

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

.....ที่ปรึกษาโครงการ  
(ดร. ขวัญนิธิ คำเมือง)

.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. ภูพงษ์ พงษ์เจริญ)

.....กรรมการ  
(ดร. สมศักดิ์ วรรณฤม)

.....กรรมการ  
(อ. สุชาดา อิศรภรณ์)

**ชื่อหัวข้อโครงการ** การจัดลำดับงานของเครื่อง โดยใช้เทคนิคการ โปรแกรมเชิงคณิตศาสตร์ และฮิวริสติกส์

**ผู้ดำเนินโครงการ** นายธนชัย เกตุเทียน รหัส 48360502  
นายพรศักดิ์ วัคสุวิงค์ รหัส 48360618

**ที่ปรึกษาโครงการ** คร. ขวัญนิธิ คำเมือง

**สาขาวิชา** วิศวกรรมอุตสาหการ

**ภาควิชา** วิศวกรรมอุตสาหการ

**ปีการศึกษา** 2552

---

### บทคัดย่อ

โครงการวิจัยนี้จัดทำขึ้นเพื่อศึกษาและพัฒนาการจัดลำดับการทำงานของเครื่องด้วยการนำเทคนิคการใช้โปรแกรมเชิงคณิตศาสตร์และวิธีฮิวริสติกส์มาประยุกต์ใช้ โดยมีจุดประสงค์เพื่อใช้แก้ปัญหาการประมวลผลที่ใช้ระยะเวลาของโปรแกรมลินโก ซึ่งในที่นี้จะมุ่งเน้นศึกษาเรื่องการใช้งาน โปรแกรมไพธอนรวมไปถึงการสร้างระบบการทำงานของโปรแกรมโดยการใช้วิธีฮิวริสติกส์มาประยุกต์ใช้การคำนวณต่างๆ รวมไปถึงการใช้งานโปรแกรมในสถานการณ์จำลองที่หลากหลาย

จากการศึกษาพบว่า การหาผลลัพธ์ของการจัดลำดับงานบนโปรแกรมไพธอนนั้นใช้เวลาในการประมวลผลที่รวดเร็วกว่าและให้ค่าที่เหมาะสมใกล้เคียงกับค่าที่ดีที่สุด ดังนั้นโปรแกรมที่ถูกสร้างขึ้นนี้จึงเป็นประโยชน์อย่างยิ่งกับสถานการณ์ที่ต้องการความรวดเร็วในการประมวลผลโดยมีระยะเวลาจำกัดรวมไปถึงงานที่ต้องการการประมวลผลที่ต้องการการประมวลผลที่รวดเร็วในการจัดลำดับงานของเครื่องเป็นอย่างยิ่ง

**Project title** Crane scheduling by mathematical programming and heuristics

**Name** Mr. Thanachai Kettian ID. 48360502  
Mr. Pornsak Wadsuriwong ID. 48360618

**Project advisor** Dr. kwanniti Khammuang

**Major** Industrial Engineering

**Department** Industrial Engineering

**Academic year** 2009

.....

### Abstract

This research project aims to study and develop the arrangement of Crane's working by using Mathematics' program and the method of Heuristics. The objective was to get rid of the problems which was found when using Lingo's program that makes us lost many time while waiting for the process' data. This research project focuses on how to use Python's program. Furthermore, we also try to create the system of this program by using Heuristics's method to apply with the other programs especially in the program while we face with various imitate situations.

The result from this research project show us that the latency of Python's program is better than the other programs that we have ever used before because using Python's program make us get sharply the process' data with the exact numerical value similar to the real. Consequently, Python's program which was produced, is profitably to many works that want to get increasingly the process' data with the correct numerical value in limited time's situations particularly in the arrangement of crane's working.

## กิตติกรรมประกาศ

โครงการวิจัยเรื่อง “การจัดลำดับงานของเครน โดยใช้เทคนิคการ โปรแกรมเชิงคณิตศาสตร์ และฮิวริสติกส์” สำเร็จลุล่วงไปด้วยดี เนื่องจากความอนุเคราะห์ช่วยเหลือจากบุคคลหลายท่าน ได้แก่

1. อาจารย์ขวัญนิธิ คำเมือง ที่ปรึกษาโครงการวิจัย อาจารย์ได้ให้คำปรึกษาดีๆ และช่วยแก้ปัญหาต่างๆที่เกิดขึ้น โดยตลอด ให้คำแนะนำต่างๆ จนโครงการวิจัยนี้สำเร็จไปได้

2. อาจารย์ภูพงษ์ พงษ์เจริญ ที่ให้คำแนะนำในการทำโครงการวิจัย

3. อาจารย์กานต์ สี่วัฒนาที่ยังคง ที่ตรวจแบบรูปเล่มโครงการวิจัยนี้และคำปรึกษาต่างๆ ที่เกี่ยวกับการทำวิจัย

4. นางสาวชুমพูนุช คีจันทิก ที่เรียบเรียงคำในบคัคย่อและแปลเป็นภาษาอังกฤษให้

5. คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ได้ให้เงินสนับสนุนในการทำโครงการนี้ และสุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา อาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาการศึกษาต่างๆ แก่ข้าพเจ้า และเพื่อนๆทุกคนที่เป้นกำลังใจในการทำโครงการนี้

คณะผู้ดำเนินโครงการวิศวกรรม

นายธนชัย เกตุเทียน

นายพรศักดิ์ วัตสุวิงค์

20 มีนาคม 2553

# สารบัญ

	หน้า
ใบรับรองปริญญาโท.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตการทำโครงการ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 แผนการดำเนินงาน.....	3
1.7 รายละเอียดงบประมาณตลอดโครงการ.....	3
บทที่ 2 หลักการและทฤษฎี.....	4
2.1 การวิจัยดำเนินงาน.....	4
2.2 โปรแกรม Python.....	10
2.3 แบบจำลองใหม่ในการแก้ไขปัญหาการจัดลำดับงานของเครนโดยใช้โปรแกรม แก้ปัญหาทางคณิตศาสตร์ของเฮราวิล หนึ่งฤทัย และกนกพร.....	19
2.4 การโปรแกรมแบบฮิวริสติกส์.....	20
2.5 การจัดลำดับงานของเครน.....	24
บทที่ 3 วิธีดำเนินโครงการ.....	26

## สารบัญ (ต่อ)

หน้า

3.1	ศึกษาการใช้วิธีการแบบฮิวริสติกส์เพื่อมาประยุกต์ในการหาการทำงานของแตรน.....	26
3.2	ศึกษาโปรแกรม Python.....	26
3.3	เขียนโปรแกรม Python โดยการใช้วิธีการแบบฮิวริสติกส์มาประยุกต์.....	26
3.4	ศึกษาการโปรแกรมของสำเร็จรูป.....	27
3.5	ทำการทดลองเปรียบเทียบเวลาการประมวลผล.....	27
3.6	สรุปผลและจัดทำรายงาน.....	27
3.7	นำเสนองานวิจัย.....	27
<b>บทที่ 4</b>	<b>ผลการทดลองและวิเคราะห์.....</b>	<b>28</b>
4.1	วิธีฮิวริสติกส์ที่ได้พัฒนาขึ้นเพื่อแก้ปัญหาบน โปรแกรม Python.....	28
4.2	ปัญหางานที่ใช้ในการคำนวณ.....	50
4.3	ผลการทดสอบ.....	56
4.4	อธิบายผลการรัน โปรแกรม Python.....	61
<b>บทที่ 5</b>	<b>บทสรุปและข้อเสนอแนะ.....</b>	<b>63</b>
5.1	วิเคราะห์ผลการทดลอง.....	63
5.2	สรุปผลการทดลอง.....	66
	เอกสารอ้างอิง.....	67
	ภาคผนวก.....	68

## สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแสดงแผนการดำเนินงาน.....	3
4.1 ปัญหางานในขนาดและลักษณะต่าง ๆ.....	54
4.2 ปัญหางานของกลุ่มวิจัยของเอราวิลและคณะ.....	55
4.3 ตารางแสดงผลลัพธ์การประมวลผลปัญหางานที่สุ่มขึ้นใหม่ของแบบจำลองอิวิริสติกส์บนโปรแกรม Python.....	56
4.4 ตารางแสดงผลลัพธ์การประมวลผลปัญหางานตามวิจัยเอราวิลและคณะของแบบจำลองอิวิริสติกส์บนโปรแกรม Python.....	57
4.5 ตารางแสดงผลลัพธ์การประมวลผลปัญหางานที่สุ่มขึ้นใหม่ของแบบจำลองอิวิริสติกส์บนโปรแกรมสำเร็จรูป.....	58
4.6 ตารางแสดงผลลัพธ์การประมวลผลปัญหางานตามวิจัยเอราวิลและคณะของแบบจำลองอิวิริสติกส์บนโปรแกรมสำเร็จรูป.....	59
5.1 ผลเปรียบเทียบการทำงานของการทำงานของการประมวลผลบนโปรแกรมสำเร็จรูป กับ โปรแกรม Python โดยใช้ปัญหาของเอราวิลและคณะ.....	63
5.2 ผลเปรียบเทียบการทำงานของการทำงานของการประมวลผลบนโปรแกรมสำเร็จรูป กับ โปรแกรม Python โดยใช้ปัญหาที่สร้างขึ้น.....	64



## สารบัญรูป

รูปที่	หน้า
4.1 แผนผังการทำงานของโปรแกรม Greedy 1.....	29
4.2 แสดงการทำงานในแต่ละกรณีของ Greedy 1 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	30
4.3 แผนผังการทำงานของโปรแกรม Greedy 2.....	32
4.4 แสดงการทำงานในแต่ละกรณีของ Greedy 2 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	33
4.5 แผนผังการทำงานของโปรแกรม Greedy 3.....	35
4.6 แสดงการทำงานในแต่ละกรณีของ Greedy 3 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	36
4.7 แผนผังการทำงานของโปรแกรม Greedy 4.....	38
4.8 แสดงการทำงานในแต่ละกรณีของ Greedy 4 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	39
4.9 แผนผังการทำงานของโปรแกรม Greedy 5.....	41
4.10 แสดงการทำงานในแต่ละกรณีของ Greedy 5 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	42
4.11 แผนผังการทำงานของโปรแกรม Greedy 6.....	44
4.12 แสดงการทำงานในแต่ละกรณีของ Greedy 6 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	45
4.13 แผนผังการทำงานของโปรแกรม Greedy 7.....	47
4.14 แสดงการทำงานในแต่ละกรณีของ Greedy 7 โดยใช้ตัวอย่างปัญหาทางานที่ 6 ขนาดกลาง.....	48
4.15 กราฟแสดงลักษณะของงานแบบที่ 1.....	49
4.16 กราฟแสดงลักษณะของงานแบบที่ 2.....	50
4.17 กราฟแสดงลักษณะของงานแบบที่ 3.....	50
4.18 กราฟแสดงลักษณะของงานแบบที่ 4.....	51
4.19 กราฟแสดงลักษณะของงานแบบที่ 5.....	51
4.20 กราฟแสดงลักษณะของงานแบบที่ 6.....	52
4.21 กราฟแสดงลักษณะของงานแบบที่ 7.....	52
4.22 กราฟแสดงลักษณะของงานแบบที่ 8.....	53
4.23 กราฟแสดงลักษณะของงานแบบที่ 9.....	53

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันการขนส่งสินค้าทางเรือเป็นสิ่งสำคัญและได้รับความนิยมอย่างแพร่หลาย เพราะสามารถขนส่งสินค้าได้ในปริมาณมากต่อเรือ 1 ลำ แต่ในทางตรงกันข้ามในขณะที่การขนส่งทางเรือมีมากขึ้นเรื่อยๆ แต่ท่าเรือยังมีไม่เพียงพอกับจำนวนการขนถ่ายสินค้า เนื่องจากมีพื้นที่และอุปกรณ์ในการขนย้ายมีจำกัด ซึ่งในการขนย้ายนี้มีเครนเป็นอุปกรณ์ในการขนถ่ายสินค้าจากเรือลงสู่พื้นดิน จากปัญหาข้างต้นทำให้เกิดการรอคอยของสินค้าในการลำเลียงเป็นอย่างมาก ทำให้เกิดการสูญเสียเวลาและค่าใช้จ่ายที่เกิดจากการรอคอย จึงต้องมีการศึกษาการจัดลำดับการทำงานของเครนเพื่อจะทำให้การขนถ่ายสินค้าจากเรือเร็วขึ้น บนพื้นฐานทรัพยากรที่มีอยู่อย่างจำกัด ซึ่งก็คือพื้นที่ของท่าเรือและจำนวนของเครน

จากข้างต้น ปัจจุบันท่าเรือหลายแห่งต้องการระบบการจัดการที่มีประสิทธิภาพ ดังเช่นท่าเรือสิงคโปร์ซึ่งเป็นท่าเรือที่มีการขนถ่ายสินค้าจำนวนมากในแต่ละวันเป็นอันดับต้นๆ ของโลก ท่าเรือสิงคโปร์มีข้อจำกัดอยู่ 2 ด้านหลัก ๆ คือ พื้นที่ที่จำกัดเนื่องจากลักษณะภูมิประเทศที่มีที่ตั้งเป็นเกาะขนาดเล็ก สองก็คือเครนที่ใช้ในการลำเลียงตู้คอนเทนเนอร์มีจำนวนจำกัดในพื้นที่ที่จำกัด โดยเครนมีข้อจำกัดในการทำงาน คือ

- 1) ขณะที่เครนทำงานพร้อมกัน เครนจะไม่สามารถทำงานในเส้นทางที่ข้ามตัดกันได้
- 2) เครนจะไม่ทำงานอื่นก่อนที่จะทำงานของตัวเองเสร็จก่อน
- 3) เครนจะไม่หยุดทำงานจนกว่าจะทำงานทั้งหมดเสร็จสิ้น

จากข้อจำกัดข้างต้นนี้ เพื่อเป็นการลดเวลาในการทำงานของเครนให้น้อยที่สุด ทางกลุ่มผู้จัดทำได้สังเกตเห็นถึงความสำคัญในการจัดการทรัพยากรนี้ เพื่อที่จะสามารถลดเวลาในการทำงานและการประมวลผลให้น้อยลง ซึ่งวิธีการแก้ปัญหานี้ ทางผู้จัดทำได้สร้างออกมาในรูปแบบของแบบจำลองทางคณิตศาสตร์ โดยใช้โปรแกรม Python เพื่อแก้ปัญหการจัดลำดับการทำงานของเครน

### 1.2 วัตถุประสงค์ของโครงการ

1.2.1 ศึกษาการทำงานของเครน เพื่อนำมาปรับปรุงและประยุกต์ให้เกิดประสิทธิภาพสูงสุด

1.2.2 สร้างแบบจำลองการจัดลำดับการทำงานของเครน โดยใช้โปรแกรม Python เพื่อลดระยะเวลาการทำงานของเครน

1.2.3 ศึกษาและประยุกต์รูปแบบสมการคณิตศาสตร์เพื่อทำแบบจำลองการทำงานของเครน

1.2.4 ศึกษาและฝึกทักษะการใช้งานโปรแกรม Python เพื่อใช้ในการทำงาน

1.2.5 ปรับปรุงแบบจำลอง การจัดลำดับการทำงานของครอน โดยใช้วิธีฮิวริสติกส์ (Heuristics)

### 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ประยุกต์นำวิธีการแบบฮิวริสติกส์เข้ามาใช้เพื่อเกิดความรวดเร็วในการประมวลผลได้

1.3.2 สามารถนำโปรแกรมไปใช้ได้จริงในสถานการณ์ที่เร่งด่วน

1.3.3 ได้คำตอบที่เป็นไปได้และใช้งานได้จริง

### 1.4 ขอบเขตการทำโครงการ

1.4.1 ศึกษาการเขียนโปรแกรมและประยุกต์นำวิธีฮิวริสติกส์มาใช้

1.4.2 เขียนโปรแกรมและนำมาประยุกต์ใช้ในโปรแกรม Python

1.4.3 ศึกษาแบบจำลองและเปรียบเทียบแบบจำลองบนโปรแกรมสำเร็จรูป และ Python

### 1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาการสร้างและหาคำตอบแบบวิธีฮิวริสติกส์

1.5.2 ศึกษาและทดลองเขียน โปรแกรม Python

1.5.3 สร้างสมการและเขียน โปรแกรมพร้อมทั้งสรุปผล

1.5.4 เปรียบเทียบเวลาของการประมวลผลระหว่างโปรแกรมสำเร็จรูป กับ Python

1.5.5 บันทึกผลและสรุปผลของโครงการ

## 1.6 แผนการดำเนินงาน

ตารางที่ 1.1 ตารางแสดงการดำเนินงาน

ลำดับ	การดำเนินงาน	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1	ศึกษาการเขียนโปรแกรม Python		↔								
2	ทดลองการเขียนโปรแกรม Python ด้วยวิธีฮิวริสติกส์				↔						
3	ศึกษาการทำงานของโครง				↔						
4	ทดลองเขียนโปรแกรมแก้ปัญหาโครง					↔					
5	สรุปผลการเขียนโปรแกรม						↔				
6	เปรียบเทียบเวลาของการประมวลผล							↔			
7	ทดสอบและเปรียบเทียบแบบจำลองแต่ละแบบ								↔		
8	สรุปและจัดทำรายงาน									↔	
9	นำเสนอรายงาน										↔

## 1.7 รายละเอียดงบประมาณตลอดโครงการ

1. ค่าใช้จ่ายในการเดินทาง 500 บาท
  2. ค่าใช้จ่ายด้านเอกสาร 1500 บาท
- รวมเป็นเงิน 2000 บาท (สองพันบาทถ้วน)

## บทที่ 2

### หลักการและทฤษฎีเบื้องต้น

เนื้อหาในบทนี้เราจะเริ่มกล่าวถึงตั้งแต่ความเป็นมาของการวิจัยดำเนินงาน รวมไปถึงบทบาทและขั้นตอนในการปฏิบัติในการวิจัยดำเนินงาน ฮิวริสติกส์ Python และจะปิดท้ายด้วยการจัดลำดับงานของเครน เพื่อให้ทราบถึงความจำเป็นในการนำการวิจัยดำเนินงานไปใช้ในการแก้ปัญหาการขนถ่ายสินค้าด้วยเครน

#### 2.1 การวิจัยดำเนินงาน

การวิจัยดำเนินงาน บางครั้งอาจเรียกว่า การวิจัยปฏิบัติการ หรือ การวิจัยเชิงปฏิบัติการ (OR : operations research) เป็นการใช้แบบจำลองเชิงคณิตศาสตร์มาช่วยในการตัดสินใจ โดยปกติจะใช้การวิจัยดำเนินงานในการวิเคราะห์ระบบที่มีอยู่ในโลกจริงที่มีความซับซ้อน โดยมีเป้าหมายเพื่อพัฒนาเพื่อให้ได้ประสิทธิภาพที่เหมาะสมที่สุด

##### 2.1.1 ประวัติและความเป็นมาของการวิจัยดำเนินงาน

แนวความคิดของการวิจัยดำเนินงานได้เริ่มตั้งแต่ระหว่างสงครามโลกครั้งที่ 1 โทมัส เอ็ดดิสันได้รับมอบหมายจากกองทัพเรือให้ช่วยแก้ปัญหาว่า ควรจะให้สินค้าเดินไปตามเส้นทางใดจึงจะได้ผลดีที่สุด โดยทำให้ความเสียหายในการขนส่งอันจะเกิดจากเรือค้ำน้ำของฝ่ายข้าศึกอยู่ในระดับต่ำสุด

ต่อมาในปี 1917 เอ.เค. เออร์ลาง ซึ่งเป็นนักคณิตศาสตร์ชาวเดนมาร์ก ได้วิเคราะห์ปัญหาการใช้โทรศัพท์ สูตรที่เขาสร้างขึ้นนั้นยังคงใช้กันโดยทั่วไป ในการวางแผนอุปกรณ์วงจรไฟฟ้าและความเคลื่อนไหวเข้าออกตามชุมสายโทรศัพท์ เออร์ลางได้วิเคราะห์ปัญหาการใช้โทรศัพท์เพื่อจะแก้ปัญหาในการให้บริการไม่ต้องมีการคอยกันมาก และควรจะให้มีความเป็นปริมาณเท่าใดจึงจะเพียงพอที่จะให้ผู้ใช้บริการได้รับความสะดวกเต็มที่ จากการศึกษาวิเคราะห์เออร์ลางได้พบว่าการใช้โทรศัพท์ของประชาชนจะจัดเข้าอยู่ในระบบหนึ่ง คือจะใช้กันมากในตอนเช้าและตอนบ่าย และใช้น้อยในตอนกลางคืน เขาได้แก้ปัญหาในข้อนี้ด้วยการแนะนำให้ปรับปรุงบริการ โดยจัดให้ประชาชนสามารถโทรศัพท์ถึงกันได้เป็นส่วนใหญ่ในตอนเช้าและบ่าย ให้มีการคอยบ้างเป็นจำนวนน้อย ทั้งนี้เพื่อป้องกันการสิ้นเปลืองในการติดตั้งอุปกรณ์เครื่องมือมากไป เพราะในตอนกลางคืนจะมีผู้ใช้น้อย แนวความคิดนี้เองได้สร้างรากฐานของการวิจัยดำเนินงานในปัจจุบัน

ก่อนสงครามโลกครั้งที่สอง แฮแรช-ซี เลวินสัน นักวิทยาศาสตร์ธรรมชาติวิทยาได้วิเคราะห์ปัญหาของฝ่ายการจัดการ นำเอาตัวแบบคณิตศาสตร์มาใช้กับกรณีที่มีข้อมูลเป็นจำนวนมาก นอกจากนี้เขายังศึกษาเกี่ยวกับการที่ลูกค้าปฏิเสธ ไม่ยอมรับหรือพัสดุที่ส่งชื่อจากบริษัทขาย

สินค้าทางไปรษณีย์ ปรากฏว่าโดยตัวเฉลี่ยแล้วเขาปฏิเสธไม่ยอมรับสินค้าที่สั่งซื้อซึ่งมีมากกว่าร้อยละ 30 ของยอดขายขั้นต้นและมีผลกระทบกระเทือนต่อยอดกำไร เหตุผลที่ค้นพบมีทั้งที่ชัดเจนและไม่ชัดเจน เหตุที่สำคัญมีสองประการ ประการแรกคือคำสั่งซื้อที่มีจำนวนเงินสั่งซื้อยิ่งสูงเพียงใด โอกาสที่จะถูกปฏิเสธก็ยิ่งมีมากเพียงนั้น ประการที่สองเกี่ยวกับช่วงระยะเวลาระหว่างได้รับใบสั่งซื้อและการส่งสินค้า ถ้าบริษัทส่งสินค้าหลังจากได้รับใบสั่งแล้ว 5 วันจะไม่ได้ผล ทั้งนี้ต้องมีการเปรียบเทียบต้นทุนในการปฏิเสธสินค้าที่ส่งไปกับต้นทุนที่จะส่งสินค้าให้เร็วขึ้น

ต่อมาในระยะเวลาห่างสงครามโลกครั้งที่สองประมาณปี ค.ศ.1937 อังกฤษได้นำการวิจัยขั้นดำเนินการมาใช้เป็นครั้งแรกในการบริหารงานทางทหาร ในขณะนั้น Sir Robert Watson-Watt นักวิทยาศาสตร์ชาวอังกฤษเป็นหัวหน้ากลุ่มนักวิทยาศาสตร์ได้รับมอบหมายจากรัฐบาลอังกฤษทำการวิเคราะห์ปัญหาการใช้อุปกรณ์เรดาร์ในการจับเครื่องบิน และเวลาที่เครื่องบินเข้าศึกเข้าโจมตีจริงๆ ถ้าเรดาร์จับได้เร็วจะทำให้มีเวลาเตรียมการส่งเครื่องบินประจัญบานขึ้นไปต่อสู้ได้ทันการ นักวิทยาศาสตร์ได้วิเคราะห์ลักษณะการปฏิบัติการของสถานีเรดาร์แต่ละแห่งตลอดจนวงจรการสื่อสารและวิธีการปฏิบัติงานจนได้ผลเป็นตัวเลขเสนอรัฐบาล เพื่อปรับปรุงหน่วยสถานีเรดาร์ทั้งหมดของกองทัพอากาศอังกฤษ ผลงานครั้งนี้ใช้ได้ดีมากแสดงว่านักวิทยาศาสตร์สามารถช่วยนักบริหารในการวิเคราะห์แก้ไขปัญหาต่างๆ ได้ ต่อมาในปี ค.ศ.1941 กองทัพอากาศอังกฤษได้จัดตั้งหน่วยวิจัยขั้นดำเนินการทางทหาร และนำไปใช้แพร่หลายในหน่วยงานของกองทัพบกและกองทัพเรืออีกด้วย ในปี ค.ศ.1942 รัฐบาลสหรัฐก็ได้ตั้งหน่วยวิจัยขั้นดำเนินการขึ้นเป็นครั้งแรกโดยได้รับคำแนะนำจากอังกฤษและเป็นการร่วมมือกับการปฏิบัติการทางทหารของอังกฤษ ในประเทศสหรัฐอเมริกาได้ตั้งหน่วยวิจัยขั้นดำเนินการโดยใช้ชื่อต่างกัน ใช้ชื่อหน่วยวิจัยขั้นดำเนินการ (operations research) ในกองทัพบก ใช้ชื่อหน่วยประเมินขั้นดำเนินการ (operations evaluation) ในกองทัพเรือ และกองทัพอากาศใช้หน่วยวิเคราะห์ขั้นดำเนินการ (operations analysis)

หลังจากสงครามโลกครั้งที่สอง ประเทศอังกฤษก็เป็นประเทศแรกที่น่าวิธีการวิจัยขั้นดำเนินการมาใช้ในวงการอุตสาหกรรมและรัฐวิสาหกิจ อุตสาหกรรมประเภทแรกที่น่าวิชาการนี้มาใช้ในการบริหารงานคืออุตสาหกรรมทำถ่านหิน ต่อมาจึงได้ขยายตัวเข้าไปสู่อุตสาหกรรมประเภทอื่นๆ และนิยมใช้ในการบริหารงานด้านการขนส่ง ส่วนสหรัฐอเมริกานั้นไม่ค่อยให้ความสนใจในการนำไปประยุกต์กับงานทางด้านธุรกิจมากนักในระยะแรก จนกระทั่งได้มีการขยายตัวในการใช้เครื่องจักรแทนคน สหรัฐฯจึงได้ทำการฟื้นฟูและส่งเสริมหลักการนี้และนิยมใช้อย่างแพร่หลาย ได้มีการตั้งสมาคมและจัดสอนขึ้นในมหาวิทยาลัยและสถาบันการศึกษาขั้นสูงโดยทั่วไป

จากความสำเร็จของทีมนักวิทยาศาสตร์ของ ฝ่ายบริหารทางทหารของอังกฤษที่ศึกษาถึงสถานการณ์และวิกฤติการณ์ของปัญหาที่เกี่ยวกับการสู้รบ เพื่อป้องกันประเทศชาติ ทั้งทางบก ทางน้ำ และทางอากาศเพื่อหาแนวทางปฏิบัติและตัดสินใจเพื่อเลือกใช้อาวุธยุทธ โขปกรณ์ที่มีอยู่โดย

จำกัดให้เกิดผลอย่างมีประสิทธิภาพทำให้สหรัฐอเมริกาเริ่มสนใจ และนำไปดำเนินการข้างงาน ประสบผลสำเร็จอย่างมากในการแก้ไขปัญหาด้านการเกษตรซึ่งค่อนข้างซับซ้อน ปัญหาทางการสร้าง เครื่องบินแบบใหม่ๆ การวางแผนท่าเหมืองในทะเล การใช้เครื่องมือทางอิเล็กทรอนิกส์อย่างมีประสิทธิภาพตลอดถึงการจัดการทางการเกษตรกรรม

เมื่อสงครามโลกสิ้นสุดลงความสำเร็จอย่างงดงามของทีมงานทางทหารที่ใช้วิชาการวิจัยดำเนินงาน ได้มีผลทำให้ฝ่ายจัดการในวงการอุตสาหกรรมมีความสนใจและนำหลักการที่พัฒนาขึ้นมาใหม่นี้มาใช้จนสามารถแก้ไขปัญหาด้านการผลิตและด้านอื่นๆในอุตสาหกรรมได้อย่างดี ความรู้ในลักษณะวิธีการที่เรียกว่า เทคนิคทางวิชาการวิจัยดำเนินงาน มีแนวโน้มที่จะเจริญพัฒนาไปอย่างกว้างขวางมาก ทั้งนี้เพื่อสนองความต้องการในการแก้ไขปัญหาด้านที่ซับซ้อน ขนาด และความซับซ้อนตามการขยายตัวของสังคมและเศรษฐกิจ ตัวอย่างที่พบเห็นได้ง่ายก็คือ การแบ่งแยกงานตามความชำนาญของแรงงานเป็นการเพิ่มความรับผิดชอบให้ฝ่ายจัดการมากขึ้น ในการที่จะต้องแบ่งความรับผิดชอบ ความชำนาญและการจัดสรรองค์ประกอบที่มีอยู่ให้เหมาะสมกับงานยิ่งขึ้น และยังคงแก้ไขปัญหาดังกล่าวอย่างต่อเนื่องอีกด้วย การพัฒนาทางวิชาการวิจัยดำเนินงานจึงเป็นไปอย่างรวดเร็ว

### 2.1.2 บทบาทของการวิจัยดำเนินงาน

การวิจัยดำเนินงานเป็นงานส่วนที่มีบทบาทต่อการบริหารขององค์กรอย่างมาก บทบาทที่สำคัญก็คือ การวางแผน เสนอแนวปฏิบัติและการควบคุมการดำเนินงานขององค์กร การปรับปรุงประสิทธิภาพการทำงานและแก้ไขปัญหาด้านหน่วยงานต่างๆ ตัวอย่างองค์กรหรือหน่วยงานที่ใช้วิธีการทางวิจัยดำเนินงาน มีดังนี้

#### 2.1.2.1 ทางทหาร

- 1) ช่วยปรับปรุงระบบสื่อสาร และ การใช้เครื่องมือทางการสื่อสาร
- 2) ใช้กับการวางแผนงานทางยุทธวิธี เพื่อกำหนดความต้องการและจัดสรรอาวุธ ยุทโธปกรณ์และกำลังพลอย่างมีประสิทธิภาพ
- 3) แก้ปัญหาทางด้านการผลิตอาวุธ ฝึกทหาร เลือจุดยุทธศาสตร์ เลือกแนวเส้นทางกำลังพลและสัมภาระ ฯลฯ

#### 2.1.2.2 ทางอุตสาหกรรม

- 1) แก้ปัญหาการจัดสถานที่ตั้งโรงงาน คลังสินค้า ให้เหมาะสมเพื่อลดค่าใช้จ่าย
- 2) ช่วยในการวางแผนงานจัดกำลังคน เครื่องจักร วัสดุ และกำหนดช่วงเวลาการผลิตให้เหมาะสมเป็นการลดค่าใช้จ่ายและเพิ่มการผลิต
- 3) ช่วยให้สามารถเลือกใช้วัสดุ เครื่องจักร และกระบวนการที่ถูกต้อง

### 2.1.2.3 ทางสาธารณสุขโรคอื่นๆ

1) โรงพยาบาล ใช้ช่วยในการจัดเตรียมจำนวนคงคลังของยา และจัดรูปการบริการของการรักษาพยาบาล

2) ระบบขนส่งและการจราจร ใช้ช่วยในการวางแผนเพื่อให้ระบบดีขึ้น

3) กิจการ โรงแรมช่วยให้เกิดการบริการที่ดีขึ้น

4) การก่อสร้าง ใช้ประโยชน์ในการวางแผนและควบคุมงาน

5) ห้องสมุดใช้ประโยชน์ในการบริหารและบริการ เช่น การจัดสถานที่วางหนังสือ การจัดสถานที่อ่านหนังสือ และการป้องกันหนังสือหาย

6) ทางกรมตำรวจสามารถช่วยจัดสถานที่ตั้งสถานีตำรวจ และสถานีดับเพลิง

7) องค์การโทรศัพท์ ใช้ในการจัดตู้สาย และระบบของหมายเลข

8) การประปา ใช้ในการปรับปรุงระบบส่งน้ำ

### 2.1.2.4 ทางธุรกิจต่างๆ

1) สถาบันการเงินต่างๆ สามารถใช้ในการจัดสรรเงินกู้และบริหารการเงิน

2) บริษัทดำเนินธุรกิจ ใช้ในการช่วยตัดสินใจสั่งซื้อ การขาย และการควบคุม

สินเชื่อ

3) ชาวไร่ชาวนาสามารถใช้เพื่อการวางแผนการเพาะปลูก

### 2.1.3 ขั้นตอนการศึกษาปัญหาในการวิจัยดำเนินงาน

การวิจัยดำเนินงานเน้นเทคนิคในเชิงปริมาณ (QUANTITATIVE TECHNIQUE) อย่างมาก ฉะนั้นการตั้งรูปแบบของสมการหรืออสมการในทางคณิตศาสตร์แทนระบบของปัญหาและอาศัยความรู้ทางคณิตศาสตร์มาช่วยในการวิเคราะห์ปัญหานั้นๆจึงเป็นงานของกลุ่มทีมงานด้านการวิจัยดำเนินงานโดยตรง ซึ่งโดยหลักทั่วไปแล้วขั้นตอนในการดำเนินงาน โดยสรุปมีดังนี้

#### 2.1.3.1 การจัดตั้งปัญหา (FORMULATING THE PROBLEM)

เนื่องจากปัญหาต่างๆอยู่ในลักษณะที่ไม่ชัดเจนและมีขอบข่ายกว้างเกินไป ยากแก่การเข้าใจและแก้ไขให้ได้ผลลัพธ์ถูกต้องแน่นอน การตั้งปัญหาให้ถูกต้องและเข้าใจง่ายจะช่วยให้การหาผลลัพธ์ง่ายขึ้นมาก การตั้งปัญหาจึงมีความสำคัญอย่างยิ่ง ขั้นตอนในการจัดตั้งปัญหามีดังนี้

1) ศึกษาระบบที่กำลังเป็นปัญหาและความสัมพันธ์อื่นๆ ที่เกี่ยวข้อง

2) กำหนดปัญหา และเลือกแบ่งแยกปัญหา โดยกำหนดข้อความของปัญหา

อย่างรัดกุมและเข้าใจง่าย

3) กำหนดจุดประสงค์และเลือกวิธีการจัดการผลการดำเนินงาน

4) กำหนดขอบข่ายของปัญหาและข้อสมมุติต่างๆ



- 5) กำหนดแนวทางดำเนินงานต่างๆ ที่เป็นไปได้เพื่อการแก้ปัญหา
- 6) กำหนดช่วงเวลาที่ใช้ในการวิเคราะห์ปัญหาและในการตัดสินใจเพื่อดำเนินงาน ในการแก้ปัญหาอื่นๆ

### 2.1.3.2 การสร้างรูปแบบทางคณิตศาสตร์แทนระบบของปัญหา (CONSTRUCTING - A MATHEMATICAL MODEL)

เมื่อกำหนดและเข้าใจปัญหาอย่างถูกต้องและมีจุดประสงค์สอดคล้องกัน เพื่อหาแนวทางดำเนินงานเป็นที่ถูกต้องแล้ว งานขั้นต่อไปคือ การจัดรูปแบบของปัญหาให้ง่ายต่อการวิเคราะห์ยิ่งขึ้น การจัดรูปแบบของปัญหาโดยทั่วไปนิยมใช้อยู่ 2 วิธี คือ การสร้างรูปแบบทางคณิตศาสตร์ และการจัดรูปแบบโดยใช้รูปภาพ แผนภูมิ กราฟ เข้ามาช่วย แต่ในที่นี้จะกล่าวถึงแบบทางคณิตศาสตร์สำหรับแทนระบบของปัญหา โดยมีสมการต่างๆ แสดงความสัมพันธ์ขององค์ประกอบต่างๆที่เกี่ยวข้องกัน รูปแบบทางคณิตศาสตร์ที่ใช้แทนระบบของปัญหาจะมีโครงสร้างดังนี้

- 1) มีสมการหรือฟังก์ชันของเป้าหมาย (OBJECTIVE FUNCTION)
- 2) มีตัวแปรซึ่งควบคุมได้ (DECISION VARIABLE)
- 3) มีเงื่อนไข (CONSTRAINTS)

รูปแบบทางคณิตศาสตร์นี้ ความพยายามในการแก้ปัญหาให้เป็นไปตามจุดประสงค์สามารถทำได้โดยการวิเคราะห์สมการ เปลี่ยนแปลงค่าตัวแปรภายใต้เงื่อนไขที่กำหนด และเลือกแนวทางการดำเนินงานที่มีผลดีเหมาะสมที่สุด

### 2.1.3.3 การหาผลลัพธ์ของปัญหา (DERIVING A SOLUTION)

การหาแนวปฏิบัติที่เหมาะสมที่สุดเป็นแนวความคิดที่ถูกต้องกว่า การที่จะได้มาซึ่งผลลัพธ์ที่ถูกต้องจำเป็นต้องอาศัยวิธีการทางคณิตศาสตร์ และการวิจัยดำเนินงานมาช่วยวิเคราะห์รูปแบบที่สร้างขึ้นแทนระบบของปัญหา โดยพยายามหาผลลัพธ์ที่สอดคล้องกับจุดมุ่งหมายของการแก้ปัญหาในแนวทางที่เหมาะสมกับสภาพการณ์แทนที่จะมุ่งเอาผลประโยชน์สูงสุด ซึ่งแนวทางในการได้ผลลัพธ์ที่เหมาะสมที่สุดย่อมมีได้หลายๆ แนวทาง นั่นคือ ในการที่จะได้ผลลัพธ์อย่างเดียวกัน เราอาจมีวิธีดำเนินงานแตกต่างกันได้ สิ่งที่น่าพึงระวังในการวิเคราะห์หาผลลัพธ์ที่เหมาะสมคือ การได้ผลลัพธ์ที่มีลักษณะเกิดผลดีเพียงบางส่วน ซึ่งในทางปฏิบัติสำหรับการวิจัยดำเนินงานถือว่าเป็นอันตรายมาก ทั้งนี้มีเหตุผลว่าความพยายามในการให้ผลดีเพียงส่วนหนึ่งของปัญหาจะมีผลทำให้ไม่ได้ผลลัพธ์ที่ดีสำหรับทั้งระบบ

#### 2.1.3.4 การทดสอบรูปแบบแทนระบบและผลลัพธ์ (TESTING THE MODEL AND SOLUTION)

เนื่องจากความบกพร่องในการละเว้นองค์ประกอบที่สำคัญบางส่วน ซึ่งหมายความว่าถึงรูปแบบที่เรานำมาวิเคราะห์ให้ไม่ใช่ตัวแทนระบบของปัญหาที่ดี การหาผลลัพธ์ที่ดีจึงเป็นไปได้ การทดสอบรูปแบบแทนระบบจึงเป็นส่วนที่ขาดไม่ได้ วิธีดำเนินงานตรวจสอบข้อบกพร่องของรูปแบบแทนระบบอาจทำได้โดย

- 1) ตรวจสอบวิธีการจัดตั้งปัญหาใหม่ แล้วเปรียบเทียบกับรูปแบบของปัญหา
- 2) การตรวจหน่วยที่ใช้สำหรับตัวแปรต่างๆในระบบสมการให้ถูกต้องตามสูตร
- 3) การเปลี่ยนแปลงค่าตัวแปร ถ้าค่าผลลัพธ์ยังอยู่ภายในเงื่อนไขที่ถูกต้องได้ ก็แสดงว่ารูปแบบแทนระบบนั้นใช้งานได้
- 4) การใช้ข้อมูลในอดีตทดสอบรูปแบบแทนระบบ ได้ผลอย่างไรจึงนำมาเปรียบเทียบกับความเป็นจริงที่เกิดขึ้น

#### 2.1.3.5 การตั้งขอบข่ายการควบคุมผลลัพธ์ (ESTABLISHING CONTROL OVER THE SOLUTION)

สถานะและเงื่อนไขของสิ่งแวดล้อมของระบบปัญหา มักจะเปลี่ยนไปตามกาลเวลา โดยมีผลทำให้รูปแบบแทนระบบบิดเบือนไปด้วย จึงให้ผลลัพธ์ที่ผิดพลาดไป ก็มีความจำเป็นอย่างยิ่งในการแก้ไขข้อบกพร่องในรูปแบบแทนระบบ เพื่อจะได้ผลลัพธ์ที่เป็นประโยชน์ยิ่งขึ้น การควบคุมผลลัพธ์ให้อยู่ในขอบข่ายที่ใกล้เคียงสภาพความเป็นจริง

#### 2.1.3.6 การนำผลลัพธ์ไปใช้งาน (IMPLEMENTATION)

การนำผลลัพธ์ออกใช้ดำเนินการมีหลักการดังนี้

- 1) หน่วยการวิจัยดำเนินงาน ต้องสามารถชี้แจงให้ฝ่ายบริหารเข้าใจถึงการคัดแปลง ผลที่ได้และวิธีการนำไปใช้งานให้เกิดประโยชน์ และต้องร่วมมือกันในการพัฒนาวิธีการเพื่อนำหลักการของผลลัพธ์นั้นๆ ออกใช้งาน

2) จะต้องมีการประเมินผลการดำเนินงาน เพื่อช่วยให้ฝ่ายบริหารและทีมงานการวิจัยดำเนินงาน สามารถติดตามผลการดำเนินงาน จนกระทั่งพบข้อบกพร่องและแก้ไขให้ทันเวลา

## 2.2 โปรแกรม Python

ไพทอน (Python) เป็นภาษาโปรแกรมในลักษณะภาษาอินเทอร์พรีเตอร์โปรแกรมมิ่ง (Interpreted programming language) ผู้คิดค้นคือ Guido van Rossum ในปี 1990 ซึ่งไพทอนเป็นการจัดการชนิดของตัวแปรแบบแปรผันตามข้อมูลที่บรรจุอยู่ (Fully dynamically typed) และใช้การจัดการหน่วยความจำเป็นอัตโนมัติ (Automatic memory management) โดยได้เป็นการพัฒนาและผสมผสานของภาษาอื่น ๆ ได้แก่ ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl และภาษาไพทอนยังเป็นแนวคิดที่ทำให้เกิดภาษาใหม่ ๆ ซึ่งได้แก่ Ruby และ Boo เป็นต้น ไพทอนนั้นพัฒนาเป็นโครงการ Open source โดยมีการจัดการแบบไม่หวังผลกำไรโดย Python Software Foundation และสามารถหาข้อมูลและตัวแปรภาษาได้จากเว็บไซต์ของไพทอนเอง ที่ <http://www.python.org/>

### 2.2.1 การแสดงผลเบื้องต้น (Printing)

การแสดงผลออกทางหน้าจอของไพทอนนั้นใช้คำสั่ง print แล้วตามด้วย 'String' หรือ "String" โดยแทนที่ String ด้วยข้อความใด ๆ เช่น

\* ในภาษา C/C++, Java, ฯลฯ เครื่องหมายที่บอกการจบ ของ คำสั่งคือ ; (Semi-colon Symbol) แต่ Python ใช้การจบบรรทัดแทน

### 2.2.2 การตั้งตัวชื่อแปร และคำสงวน (Reserved word หรือ Keywords)

การตั้งตัวชื่อแปร มีดังนี้

- 1) ขึ้นต้นด้วยตัวอักษรในภาษาอังกฤษ ตามด้วยตัวอักษรหรือตัวเลขใดๆ ก็ได้
- 2) ห้ามเว้นช่องว่าง และห้ามใช้สัญลักษณ์พิเศษนอกเหนือจาก underscore ( ) เท่านั้น
- 3) ตัวอักษรของชื่อจะคำนึงถึงความแตกต่างระหว่างอักษรตัวพิมพ์ใหญ่กับตัวพิมพ์เล็ก
- 4) การตั้งชื่อมีชื่อที่หวังว่า จะต้องไม่ซ้ำกับคำสงวน(Reserved words, Keywords)
- 5) ควรจะตั้งชื่อ โดยให้ชื่อนั้นมีสื่อความหมายให้เข้ากับข้อมูล สามารถอ่านและเข้าใจได้

คำสงวนในการใช้งาน (Reserved words, Keywords) มีดังนี้

and, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, yield, as (ไพทอน 2.5) และ with (ไพทอน 2.5)

\* อ้างอิงจากเอกสารไพทอน รุ่น 2.4.3 และ 2.5 (Python Reference Manual Release 2.4.3, documentation updated on 29 March 2006 and Python Reference Manual Release 2.5, documentation updated on 19 September 2006)

## 2.2.3 การคำนวณทางคณิตศาสตร์ (Arithmetic Mathematics)

### 2.2.3.1 การคำนวณพื้นฐาน (normal arithmetic operators)

ไพธอนนั้นมีสัญลักษณ์การคำนวณพื้นฐาน (normal arithmetic operators) ซึ่งได้แก่

- 1) \*\* (ยกกำลัง, Exponentiation)
- 2) \* (คูณ, multiplication)
- 3) / (หาร, division)
- 4) % (หารเอาเศษ, remainder หรือ modulo)
- 5) + (บวก, addition)
- 6) - (ลบ, subtraction)

โดยอันดับความสำคัญของการคำนวณเหมือนกับคณิตศาสตร์โดยมีความสำคัญดังต่อไปนี้

- 1) วงเล็บ (parentheses "()")
- 2) ยกกำลัง (exponents "\*\*")
- 3) คูณ (multiplication "\*"), หาร (division "/") และหารเอาเศษ (remainder/modulo "%")
- 4) บวก (addition "+") และ ลบ (subtraction "-")

### 2.2.3.2 การคำนวณผ่านฟังก์ชันภายใน (Built-in Math Functions)

#### 1) การหาค่าสัมบูรณ์ (absolute value)

มีรูปแบบ Function คือ `abs(var)` เป็น function ที่ใช้หาค่าสัมบูรณ์ โดยที่ค่า `var` เป็นตัวแปรหรือจำนวนที่ต้องการหาค่า

#### 2) จำนวนที่น้อยที่สุด และมากที่สุดในกลุ่ม (smallest or largest values)

ฟังก์ชันหาจำนวนที่น้อยที่สุด มีรูปแบบฟังก์ชัน คือ `min(var)` เป็นฟังก์ชันที่ใช้ในการหาจำนวนที่น้อยที่สุดในกลุ่มของชุดข้อมูลชนิดต่าง ๆ เช่น list, set หรือแม้แต่ตัวเลขทั่วไป โดยที่ค่า `var` เป็นตัวแปร, จำนวนหรือชุดของจำนวนที่ต้องการหาค่า

ฟังก์ชันหาจำนวนที่มากที่สุด มีรูปแบบฟังก์ชัน คือ `max(var)` เป็นฟังก์ชันที่ใช้ในการหาจำนวนที่มากที่สุดในกลุ่มของชุดข้อมูลชนิดต่าง ๆ เช่น list, set หรือแม้แต่ตัวเลขทั่วไป โดยที่ค่า `var` เป็นตัวแปร, จำนวนหรือชุดของจำนวนที่ต้องการหาค่า

### 2.2.3.3 กำหนดจำนวนตัวเลขทศนิยม (specified number of digits)

มีรูปแบบฟังก์ชัน คือ `round(var, digits)` เป็นฟังก์ชันที่ใช้ในการเพิ่มหรือลดจำนวนตัวเลขทศนิยมที่จะนำมาแสดง โดย

```
>>> print round(1234.56789, 2)
```

1234.57

### 2.2.3.4 หาผลรวมทั้งหมดในชุดข้อมูล (adds numbers in a sequence.)

มีรูปแบบฟังก์ชัน คือ `sum(sequence)` เป็นฟังก์ชันที่ใช้ในการหาผลรวมของชุดข้อมูลตัวเลขหนึ่ง ๆ โดยที่ใส่ค่าของชุดจำนวนเข้าไปในรูปแบบของ `sequence of numbers` เช่น `(1,2,3,4,5)` ซึ่งหมายถึงจำนวนตั้งแต่ 1- 5 เป็นต้น ลงไปในฟังก์ชัน `sum` เพื่อทำการคำนวณหาผลรวมของชุดข้อมูลดังกล่าว

### 2.2.3.5 ช่วงของข้อมูลตัวเลข (range of numbers.)

มีรูปแบบฟังก์ชัน คือ `range(start, end [,step])` เป็นฟังก์ชันที่ใช้ในการส่งค่าช่วงของข้อมูลตัวเลขที่ต้องการออกมาเช่นต้องการตัวเลขตั้งแต่ 1 - 500 เราสามารถทำได้

### 2.2.4 ชนิดของตัวแปร (Data type)

ดังที่เราได้ทราบไปแล้วว่าภาษาไพธอนเป็นภาษาที่เป็น Interpreter Programming ซึ่งเราไม่จำเป็นต้องสนใจ Data type แต่บางครั้งการที่เรารู้จัก Data type ต่าง ๆ และนำมาใช้งานได้อย่างเหมาะสมทำให้การเขียนโปรแกรมมีประสิทธิภาพมากขึ้น โดย Data type ต่าง ๆ นั้นมีจุดเด่นในรูปแบบชนิดของตัวเองซึ่งในการแสดงผลการทำงานของข้อมูลต่าง ๆ ในไพธอนนั้นมีหลากหลายรูปแบบ เราจะมาพูดคุยกันในเรื่องนี้เช่นเดียวกัน โดยในหลาย ๆ ภาษานั้น ๆ เราจำเป็นต้องประกาศชนิดตัวแปร (Data type) ก่อน แล้วจึงตั้งชื่อตัวแปร ซึ่งหลาย ๆ ครั้งสร้างความสับสนในการจดจำแต่ในภาษาไพธอนนั้นเราไม่จำเป็นต้องประกาศชนิดของตัวแปรก่อนการใช้งานแต่อย่างใด โดยในบทนี้เราจะพูดถึงชนิดของตัวแปรต่าง ๆ

#### 2.2.4.1 ตัวเลข (Numbers)

##### 1) จำนวนเต็ม(Integers)

- **จำนวนเต็มธรรมดา (Plain Integer)** Plain Integers มีคำย่อสำหรับเขียนในโปรแกรมคือ `int` เป็นการบ่งบอกคุณสมบัติของตัวแปรที่ใช้เก็บข้อมูลตัวเลขแบบจำนวนเต็มที่เป็นตัวเลขแบบ signed integer เก็บข้อมูล 32 bits ตั้งแต่ -2147483648 ถึง +214748364

- **จำนวนเต็มแบบยาว (Long Integer)** Long integers มีคำย่อสำหรับเขียนในโปรแกรมคือ `long` เป็นการบ่งบอกคุณสมบัติของตัวแปรที่ใช้เก็บข้อมูลตัวเลขแบบจำนวนเต็มที่เป็นตัวเลขแบบ signed integer เก็บข้อมูลที่มีมากกว่าตัวเลขที่ Integer เก็บได้หรือมีจำนวนที่มากกว่า -2147483647 ถึง + 2147483647 นั้นเอง โดยสามารถใช้คำตัวอักษร L ต่อท้ายตัวเลขนั้น ๆ เพื่อบอกว่าจำนวนนั้นเป็นจำนวน Long Integer เช่น 234187626348292917L หรือ 7L

### 2.2.4.2 จำนวนตรรกะ (Boolean)

Boolean มีคำย่อสำหรับเขียนในโปรแกรมคือ bool ความจริงแล้วข้อมูลชนิดตรรกะในไพธอนไม่มีอยู่จริงเพราะไม่จำเป็นต้องมีนั่นเอง สิ่งที่ไพธอนมีให้นั่นเป็นเพียงค่าคงที่เพื่อใช้แทนตรรกะ จริง และ เท็จ เท่านั้น ได้แก่

- 1) True - แทนค่าจริง มีค่าเป็น 1
- 2) False - แทนค่าเท็จ มีค่าเป็น 0

### 2.2.4.3 จำนวนจริง (Floating-point numbers)

Floating-point มีคำย่อสำหรับเขียนในโปรแกรมคือ float เป็นการบ่งบอกคุณสมบัติของตัวแปรที่ใช้เก็บข้อมูลตัวเลขแบบจำนวนจริงแบบ 64 bit double precision ที่มีความหมายรวบรวมกันเลยทีเดียว float และ double ในหลายๆ ภาษา แต่ในไพธอนจะใช้ float เพียงอย่างเดียว ซึ่งมีช่วงตั้งแต่ 1.23 ไปจนถึง  $7.8 \times 10^{-28}$

### 2.2.4.4 จำนวนเชิงซ้อน (Complex Numbers)

มีคำย่อสำหรับเขียนในโปรแกรมคือ complex จำนวนเชิงซ้อนคือเซตที่ต่อเติมจากเซตของจำนวนจริงโดยเพิ่มจำนวน  $j$  เข้าไปในตัวเลขจำนวนจริงจนได้เป็น จำนวนจินตภาพ (imaginary number) จนทำให้สมการ  $j^2 + 1 = 0$  เป็นจริง และหลังจากนั้นเพิ่มสมาชิกตัวอื่นๆ เข้าไปจนกระทั่งเซตที่ได้ใหม่มีสมบัติปิดภายใต้การบวกและการคูณ จำนวนเชิงซ้อน  $z$  ทุกตัวสามารถเขียนอยู่ในรูป  $x + iy$  โดยที่  $x$  และ  $y$  เป็นจำนวนจริง โดยเราเรียก  $x$  และ  $y$  ว่าส่วนจริงและส่วนจินตภาพของ  $z$  ตามลำดับ

## 2.2.5 ชนิดข้อมูลแบบการรวมกลุ่มข้อมูล (Collection Data Types)

### 2.2.5.1 ลิสต์ (List)

ในไพธอนนั้นชนิดตัวแปรที่ถูกนำมารวมกันอยู่ในชื่อเดียวกับเลขคือ อาร์เรย์ (Array) และ ลิสต์ (List) แต่ถ้านักพัฒนาต้องการใช้ อาร์เรย์แบบที่คุ้นเคยจริงๆ จำเป็นต้อง import module array ของไพธอนเข้ามาซึ่งยุ่งยากและทำงานช้ากว่าที่ควรจะเป็นลิสต์มีคำย่อสำหรับเขียนในโปรแกรมคือ list การทำงานของลิสต์นั้นเป็นการนำข้อมูลหลายๆ ชนิดมาเรียงต่อกันในลิสต์ของตัวแปรนั้น ๆ ซึ่งแตกต่างจากอาร์เรย์ที่ข้อมูลที่นำมาเรียงต่อกันต้องเป็นชนิดเดียวกันและนี่คงเป็นสาเหตุที่ทำให้อาร์เรย์ ถูกตัดทิ้งไปจากชนิดของตัวแปรพื้นฐานของไพธอนเพราะด้วยเหตุผลด้านความยืดหยุ่นของชนิดของตัวแปรนั่นเอง โดยลิสต์นั้นจะมีข้อมูลเรียงกันหลายๆ ตัว ครอบด้วยเครื่องหมาย square brackets "[" และ "]" เช่น

### 2.2.5.2 ดิกชันนารี (Dictionary หรือ Groupings of Data Indexed by Name)

ดิกชันนารี (Dictionary หรือ Groupings of Data Indexed by Name) มีคำย่อสำหรับเขียนในโปรแกรมคือ dict เป็นอนุกรมอีกอันหนึ่ง มีสภาพเหมือนอาร์เรย์ ที่มีสมาชิกทั้ง keys และ value โดยที่ นั้นจะใช้ชื่ออ้างอิงสมาชิก (associated key) แทนการใช้หมายเลขสมาชิก (index key) ซึ่งจะซ้ำกันไม่ได้ (ถ้ากำหนดค่าซ้ำ มันจะลบค่าเก่า และใช้ค่าใหม่แทน) โดยใช้การจัดเรียงข้อมูลและแก้ไขค่าไม่ได้ (จึงสามารถใช้ tuple เป็น keys ได้ ถ้าชนิดของข้อมูลสมาชิกเป็นชนิดเดียวกัน แต่ใช้ list เป็น keys ไม่ได้) แต่ลบ keys:value ได้ เวลาเรียกข้อมูล value จะค้นจาก keys ถ้าค้นไม่พบจะเกิดข้อผิดพลาด

### 2.2.5.3 ทับเบิล (Tuples) และ อนุกรม (Sequences)

ทับเบิล (น้องลิสต์, Tuples) มีคำย่อสำหรับเขียนในโปรแกรมคือ tuple ข้อมูลจะอยู่ภายในวงเล็บ () แต่ตอนกำหนดค่า ถ้าอยู่โดดๆ อาจไม่ใส่วงเล็บก็ได้ โดยใช้หลักคล้ายกับลิสต์ (แต่ไม่เหมือนกันทั้งหมด) เพียงแต่เราสามารถนำลิสต์หลาย ๆ ลิสต์มาบรรจุลงในลิสต์เดียวกันเองได้ โดยไม่ต้องแยกตัวแปร

### 2.2.5.4 เซ็ต (Sets)

เซ็ต เปรียบเสมือนส่วนขยายของลิสต์ (และสตริงด้วย) และจะไม่มีสมาชิกที่มีค่าซ้ำกัน ใช้ประโยชน์เหมือนกับ เรื่องเซ็ตในวิชาคณิตศาสตร์ มีคำย่อสำหรับเขียนในโปรแกรมคือ set แต่การที่จะได้ชนิดข้อมูลแบบเซตนั้นต้องทำผ่าน function set เพื่อให้ได้เซตออกมา ซึ่งส่วนมากแล้วจะใช้ List มาทำการแปลงชนิดตัวแปรเป็นเซตอีกที ดังตัวอย่างด้านล่าง \* function set(var) เป็น function ที่ใช้ในการแปลงชนิดของข้อมูลของตัวแปรอื่น ๆ มาเป็นชนิดตัวแปรแบบ set โดยที่ตัวแปร var เป็นชื่อตัวแปรที่ต้องการแปลงชนิดข้อมูลของตัวแปรนั้น ๆ

### 2.2.5.5 ฟังก์ชันที่น่าสนใจเกี่ยวกับลิสต์และดิกชันนารี

การเรียกใช้ฟังก์ชันของลิสต์และดิกชันนารีนั้นทำคล้าย ๆ กับการเรียกใช้เมธอดในภาษา Java

#### 1) ฟังก์ชัน pop - [ลิสต์/ดิกชันนารี]

รูปแบบฟังก์ชัน Object.pop([key]) ในลิสต์นั้นเราไม่จำเป็นต้องกำหนดหมายเลขสมาชิก (index key) ก่อนการ pop เพราะเมธอดจะนำค่าบนสุดมาให้เรา แต่ถ้าต้องการ pop ในหมายเลขสมาชิกที่ต้องการเพียงแต่ใส่หมายเลขสมาชิกเท่านั้น แต่ในดิกชันนารีเราจำเป็นต้องกำหนดชื่ออ้างอิงสมาชิก (associated key) ก่อนการ pop เพราะมีเช่นนั้นเมธอดจะไม่สามารถนำค่าของสมาชิกนั้น ๆ มาให้เราได้

**2) ฟังก์ชัน append - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.append([object])` เป็นฟังก์ชันที่ใช้ในการเพิ่มข้อมูลสมาชิกลงไปในลิสต์โดยจะนำไปต่อท้ายลิสต์เสมอ

**3) ฟังก์ชัน insert - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.insert(index, object)` เป็นฟังก์ชันที่ใช้ในการเพิ่มข้อมูลสมาชิกลงไปในลิสต์โดยจะนำไปใส่ในลำดับสมาชิกที่เราต้องการ

**4) ฟังก์ชัน count - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.count([object])` เป็นฟังก์ชันที่ใช้ในการนับข้อมูลสมาชิกที่ต้องการค้นหา

**5) ฟังก์ชัน index - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.index(value, [ start , stop ])` เป็นฟังก์ชันที่ใช้ในการหาค่าของหมายเลขสมาชิก (index key) ว่าข้อมูลที่เราต้องการนั้นอยู่ที่หมายเลขสมาชิกที่เท่าใด

**6) ฟังก์ชัน extend - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.extend(list|dict)` เป็นฟังก์ชันที่ใช้ในเพิ่มสมาชิกทั้งหมดจากอีกลิสต์หรือดิกชันนารี จากตัวแปรอื่น ๆ มาไว้ที่ตัวแปรตั้งต้น

**7) ฟังก์ชัน remove - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.remove(value)` เป็นฟังก์ชันที่ใช้ในการลบค่าที่กำหนดไว้ในพารามิเตอร์ออกไปจากลิสต์

**8) ฟังก์ชัน sort - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.sort()` เป็นฟังก์ชันที่ใช้ในการจัดเรียงข้อมูล (sort) ภายในลิสต์โดยสามารถที่จะจัดเรียงได้ทั้งตามหมายเลขสมาชิก หรือจัดเรียงตามข้อมูลของสมาชิก

**9) ฟังก์ชัน reverse - [ลิสต์]**

รูปแบบฟังก์ชัน `Object.reverse()` เป็นฟังก์ชันที่ใช้ในการจัดเรียงข้อมูล (sort) ภายในลิสต์แบบย้อนกลับ

**10) ฟังก์ชัน clear - [ดิกชันนารี]**

รูปแบบฟังก์ชัน `Object.clear()` เป็นฟังก์ชันที่ใช้ในการลบข้อมูลภายในดิกชันนารีทั้งหมด

**11) ฟังก์ชัน get - [ดิกชันนารี]**

รูปแบบฟังก์ชัน `Object.get(key)` เป็นฟังก์ชันที่ใช้ในเรียกข้อมูลจากชื่ออ้างอิงสมาชิกภายในดิกชันนารี



**12) ฟังก์ชัน has key - [ดิกชันนารี]**

รูปแบบฟังก์ชัน Object.has key(key) เป็นฟังก์ชันที่ใช้ในการตรวจสอบชื่ออ้างอิงสมาชิก (associated key) ว่ามีอยู่หรือไม่

**13) ฟังก์ชัน items - [ดิกชันนารี]**

รูปแบบฟังก์ชัน Object.items(key) เป็นฟังก์ชันที่ใช้ในการแสดงรายการชื่ออ้างอิงสมาชิกและข้อมูลทั้งหมดในดิกชันนารี

**14) ฟังก์ชัน keys - [ดิกชันนารี]**

รูปแบบฟังก์ชัน Object.keys() เป็นฟังก์ชันที่ใช้ในการแสดงรายการชื่ออ้างอิงสมาชิกทั้งหมดในดิกชันนารี

**15) ฟังก์ชัน values - [ดิกชันนารี]**

รูปแบบฟังก์ชัน Object.values() เป็นฟังก์ชันที่ใช้ในการแสดงรายการข้อมูลทั้งหมดในดิกชันนารี

**2.2.6 สายอักขระ (String หรือ Array of Characters)**

สายอักขระ (Strings) เป็นการเรียงตัวของอักขระมาต่อกันมากกว่า 1 ตัวจนกลายเป็นเส้นสาย หรือเรียกอีกอย่างว่าลำดับของอักขระ (Array of Characters) โดยเราสามารถกำหนดค่าตัวแปรได้โดยใช้เครื่องหมาย single quotation ('....') หรือ double quotation (" .... ") ครอบอักขระ, สายอักขระ หรือแม้แต่ตัวเลข และรวมไปถึงสัญลักษณ์พิเศษต่าง ๆ โดยในไพธอนนั้นจะเก็บข้อมูลแบบ 8-bit strings หรือ Unicode objects ขึ้นอยู่กับข้อมูลที่จัดเก็บอยู่ในขณะนั้น

**2.2.7 การเปรียบเทียบ (Comparisons)**

ในภาษาไพธอนนั้นมีการเปรียบเทียบในเชิงคณิตศาสตร์ต่าง ๆ เป็นพื้นฐานอยู่แล้ว แต่บางอย่างก็มีที่ไม่เหมือนกับภาษาอื่น ๆ คือมีการเพิ่มเติมการเปรียบเทียบในด้านของกลุ่มข้อมูล และข้อมูลที่เหมือนกัน โดยได้เพิ่ม การเปรียบเทียบแบบ "in" และ "is" เข้ามาเพื่อเปรียบเทียบกับกลุ่มข้อมูล ส่วนใหญ่การใช้ "is" มักใช้ในการเปรียบเทียบด้านตัวอักขระและสายอักขระมากกว่าเพราะสื่อความหมายมากกว่าใช้ "=" ในการเปรียบเทียบ

โดยเมื่อมีการเปรียบเทียบแล้วเราจะได้ผลออกมาคือค่าทางตรรกะ หรือ Boolean value ซึ่งในภาษาไพธอนนั้นคือ True แทนด้วยเป็นจริง และ False แทนด้วยเป็นเท็จ และเรายังสามารถใช้ตัวเลขแทนค่าดังกล่าวได้ด้วย 0 คือ False และ 1 คือ True นั้นเอง โดยค่าที่ได้นั้นมาจากการทำการทดสอบทางตรรกศาสตร์ในเชิงเปรียบเทียบค่าทั้งสองข้างของข้อมูล

< คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย น้อยกว่า ค่าทางด้านขวาหรือไม่

<= คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย น้อยกว่าหรือเท่ากับ ค่าทางด้านขวา หรือไม่

- > คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย มากกว่า ค่าทางด้านขวาหรือไม่
- >= คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย มากกว่าหรือเท่ากับ ค่าทางด้านขวาหรือไม่
- = คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย เท่ากับ ค่าทางด้านขวาหรือไม่
- != คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย ไม่เท่ากับ ค่าทางด้านขวาหรือไม่
- < คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย ไม่มีทางเท่ากับ ค่าทางด้านขวาหรือไม่
- in คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมายอยู่ในกลุ่มข้อมูลในกลุ่มข้อมูลด้านขวาหรือไม่
- is คือ เปรียบเทียบค่าทางด้านซ้ายของเครื่องหมาย เหมือนกับ ค่าทางด้านขวาหรือไม่

### 2.2.8 การควบคุมทิศทางของโปรแกรม (Control flow, Flow of Control หรือ Alternately)

การควบคุมทิศทางของโปรแกรม เป็นการเขียนแบบ การทำงานของมนุษย์ เพราะในเหตุการณ์ต่าง ๆ เรามีการตัดสินใจในแบบต่าง ๆ กันออกไป เช่นเมื่อเดินทางไปเจอ 3 แยก เราต้องตัดสินใจ เลี้ยวซ้ายหรือเลี้ยวขวา โดยมีแผนที่ และป้ายบอกเส้นทางเป็นตัวกำหนด ให้เราเลี้ยวซ้ายหรือเลี้ยวขวา เป็นต้น ในการเขียน โปรแกรมก็คือ การจำลองตัวเราลงไป เพื่อจัดการกับสถานการณ์ต่างๆ ซึ่งเราต้องมีข้อมูลในการประกอบการพิจารณา ถ้าข้อมูลบอกเราแบบหนึ่ง เราก็ต้องทำแบบหนึ่ง แต่ถ้าข้อมูลบอกเราอีกอย่าง เราก็ต้องทำอีกอย่างที่แตกต่างกันออกไป ซึ่งการที่ต้องจัดการกับสถานการณ์ต่างๆ นี้ เราเรียกมันว่าการควบคุมการทำงานของ โปรแกรมนั่นเอง การควบคุมการทำงาน มี 3 รูปแบบ ได้แก่

- 1) การตัดสินใจ (Decisions, Choice หรือ Selection)
- 2) การวนทำซ้ำ (Loop หรือ Iteration)
- 3) การจัดการความผิดปกติของโปรแกรม (Error Checking)

ในการควบคุมทิศทาง ไม่ว่าจะเป็นการตัดสินใจ หรือการทำงานแบบวนซ้ำ เราจะต้องอาศัยการ พิจารณาข้อมูลที่มีอยู่ ประกอบการควบคุม ส่วนการจัดการความผิดปกติของ โปรแกรม นั้นเป็นการดักจับสิ่งที่จะเกิดขึ้น ได้จากการเขียนโปรแกรมที่ไม่ครอบคลุมการทำงาน ซึ่งอาจจะเกิดขึ้นได้ แต่ป้องกันโปรแกรมปิดตัวเองโดยฉับพลัน เราจึงใช้การจัดการความผิดปกติของ โปรแกรมมาช่วยในการดักจับสิ่งเหล่านี้

#### 2.2.8.1 การตัดสินใจ (Decisions, Choice หรือ Selection)

##### 1) if Statements

คำสั่ง if ใช้ตัดสินใจว่าจะทำหรือไม่ทำคำสั่งชุดหนึ่งที่อยู่ในช่วงของการทำงานของ if (if Statements scope) ถ้าเงื่อนไขที่นำมาทดสอบทางตรรกศาสตร์เป็นจริง (True) ก็จะทำการคำสั่งชุดนั้นถ้าเงื่อนไขที่นำมาทดสอบทางตรรกศาสตร์เป็นเท็จ (False) ก็จะไม่ได้ทำการคำสั่งชุดนั้น โดยมีรูปแบบคำสั่งในภาษาไพธอนดังนี้

## 2) switch Statements

ในไพธอนนั้นไม่สนับสนุนการตัดสินใจแบบ switch

### 2.2.8.2 การวนทำซ้ำ (Loop)

#### 1) while Statements

คำสั่ง while เป็นคำสั่งที่ใช้ในการวนทำซ้ำในช่วงของการทำงานของ while (while Statements scope) จนกว่าการทดสอบทางตรรกศาสตร์จะเป็นเท็จ

#### 2) for Statements

คำสั่ง for เป็นคำสั่งที่ใช้ในการวนทำซ้ำในช่วงของการทำงานของ for (for Statements scope) จนกว่าการทดสอบทางตรรกศาสตร์จะเป็นเท็จ โดยมีการกำหนดช่วงการทำงานต่ำสุด และมากที่สุดของจำนวนครั้งที่ทำคำสั่งภายในด้วย โดยใช้ for ร่วมกับคำสั่งในการเปรียบเทียบทางตรรกศาสตร์คือ in และ function การทำงานแบบช่วงของกลุ่มของข้อมูลคือ range

#### 3) pass, break, continue และ else Clauses Statements

pass เป็นคำสั่งที่ใช้ในการทดแทน Statements อื่น ๆ โดยไม่มีการทำงาน

break คำสั่งนี้ใช้ในการหยุดการทำงานของ การวนทำซ้ำ หรือการตัดสินใจ

continue คำสั่งนี้ใช้ในการเริ่มการทำงานหลังจากหยุดการทำงานของ การวนทำซ้ำ หรือการตัดสินใจ

else Clauses คำสั่งนี้ใช้ในการทำหลังจาก break ใน การวนทำซ้ำ (Loop, Iteration)

### 2.2.9 การสร้างฟังก์ชัน (Defined Function)

ฟังก์ชันคือแหล่งรวมชุดคำสั่งหลาย ๆ โดยคำสั่งที่เราเรียบเรียงขึ้นเอง เพื่อนำไปใช้ในการเขียนโปรแกรมซ้ำ ๆ กันโดยไม่ต้องเขียนชุดคำสั่งนั้น ๆ ใหม่อีกครั้ง (Reusability Code) โดยในไพธอนนั้นก็มีฟังก์ชันอยู่ 2 แบบคือ สามารถคืนค่ากลับมาได้ (real function, return value) และแบบไม่คืนค่า (void, sub, subprogram หรือ subroutine) การสร้างฟังก์ชันจะใช้คีย์เวิร์ดชื่อ def แล้วตามด้วยชื่อของฟังก์ชันนั้น ๆ โดยลักษณะของช่วงการทำงานเหมือนกับคำสั่งควบคุมทิศทาง โปรแกรมต่าง ๆ เช่นเดียวกัน

- 1) function name ชื่อฟังก์ชัน
- 2) Statements ชุดคำสั่ง
- 3) var ตัวแปร
- 4) Argument รับค่าของฟังก์ชัน
- 5) return การคืนค่ากลับ โดยที่การคืนค่ากลับนั้นสามารถคืนค่ากลับได้มากกว่า 1 ค่า หรือ มากกว่า 1 ตัวแปร ซึ่งสามารถคืนค่าได้ทุกชนิดข้อมูลด้วย

### 2.2.10 เรื่องที่ห้ามลืมใน Python

- 1) Don't forget the colons อย่าลืม โคลอน (Colon, :) เมื่อจบคำสั่งควบคุมทิศทางของโปรแกรมต่าง ๆ เช่น if, while หรือ for เป็นต้น
- 2) Start in column 1 ไพธอนใช้ระบบแท็บแทนเครื่องหมายปีกกาเพื่อควบคุมช่วงของการทำงาน และต้องเริ่มจากคอลัมน์ที่ 1 ทุกครั้งด้วย
- 3) Blank lines matter at the interactive prompt อย่าปล่อยให้บรรทัดว่างหรือขึ้นบรรทัดใหม่ใน shell prompt ถ้าไม่ชัวร์ว่าเขียนคำสั่งจบในบรรทัดนั้น ๆ
- 4) Indent consistently อย่าใช้ปุ่มแท็บปนกับการเคาะเว้นวรรค โดยควรเลือกว่าจะใช้การแท็บหรือเคาะวรรค ถ้าใช้เคาะวรรคก็ควรใช้ให้ตลอดรอดฝั่ง โดย 1 แท็บ ให้มีขนาดเท่ากับเคาะวรรค 4 ครั้ง (4 whitespaces/tab)
- 5) Don't code C in Python อย่าเขียนโค้ดแบบ C ในไพธอน เช่น if (X==1): print X โดยในความเป็นจริงแล้วไม่ต้องมี ( ) ก็ได้ เป็นต้น
- 6) Don't always expect a result บาง Method เช่น Append หรือ Sort อย่าไปคิดว่ามันจะ return obj บางครั้งมัน return None หรือ Null ออกมา ซึ่งเราไม่จำเป็นต้องเขียน list=list.append(X) แต่ให้เขียน list.append(X) ลงไปได้เลย
- 7) Use calls and imports properly หลังเรียก Method ให้มี () ด้วยเช่น function() อย่าใช้ function เฉยๆ และตอน Import ไม่ต้องใส่นามสกุล file อย่าง import mod.py ใช้ import mod เฉยๆ ก็พอ

### 2.3 แบบจำลองใหม่ในการแก้ไขปัญหาการจัดลำดับงานของเครนโดยใช้โปรแกรม

#### แก้ปัญหาทางคณิตศาสตร์ของเอราวิลด หนึ่งในฤทัย และกนกพร

โดยแบบจำลองใหม่นี้เป็นแบบจำลองที่ใช้แนวความคิดประยุกต์ปรับปรุงจากแบบจำลองของ Zhu และ Lim โดยทำการศึกษาถึงตัวแปร  $x_{ijm}$  กับ  $y_{ij}$  ทำให้เวลาประมวลผลนานกว่าที่เป็นจริง หลังจากการศึกษาแบบจำลองเดิมแล้วเสร็จ ได้ทำการปรับปรุงโดยทำการตัดสมการที่มีตัวแปร  $z_{ijm}$  ออกแล้วหาความสัมพันธ์ระหว่างตัวแปร  $x_{ik}$  กับ  $y_{ij}$  ขึ้นมาดังสมการที่ (6) และ (7)

กำหนดให้แบบจำลองนี้มีค่า  $M$  คือค่าคงที่ที่มีค่ามากกว่าหนึ่ง

Minimize  $c_{\max}$

$$c_{\max} \geq c_i \quad \text{โดยที่ } 1 \leq i \leq n \quad (1)$$

$$c_i - p_i \geq 0 \quad \text{โดยที่ } 1 \leq i \leq n \quad (2)$$

$$\sum_{k=1}^m x_{ik} \leq 1 \quad \text{โดยที่ } 1 \leq i \leq n \quad (3)$$

$$c_i - (c_j - p_j) + y_{ij}M \geq 0 \quad \text{โดยที่ } 1 \leq i < j \leq n \quad (4)$$

$$c_i - (c_j - p_j) - (1 - y_{ij})M \leq 0 \quad \text{โดยที่ } 1 \leq i < j \leq n \quad (5)$$

$$x_{ik} + x_{jk} \leq (y_{ij} + y_{ji})M + 1 \quad \text{โดยที่ } \forall 1 \leq i < j \leq n \text{ และ } \forall 1 \leq k \leq m \quad (6)$$

$$x_{ik} + x_{jl} \leq (y_{ij} + y_{ji})M + 1 \quad \text{โดยที่ } \forall 1 \leq i < j \leq n \text{ และ } \forall 1 \leq k < l \leq m \quad (7)$$

## 2.4 การโปรแกรมแบบฮิวริสติกส์

การพิจารณาหาทางเลือกที่เหมาะสมให้กับปัญหาการตัดสินใจที่ซับซ้อน จะเกี่ยวข้องกับข้อจำกัดด้านเวลาและค่าใช้จ่าย หรือบางครั้งอาจเป็นไปได้ยากในการนำมาปฏิบัติจริง เนื่องจากการจำลองเหตุการณ์ (simulation) อาจจะต้องใช้เวลาค่อนข้างซับซ้อนและไม่แม่นยำ บางครั้งการใช้ฮิวริสติกส์อาจทำให้ได้ผลลัพธ์ที่น่าพึงพอใจได้เร็วกว่า และเสียค่าใช้จ่ายน้อยกว่า เนื่องจากเป็นวิธีที่ทำการทดลองค้นคว้าหากฎด้วยตัวเอง ใช้คุณพินิจหรือวิจาร์ญาณในการเลือก ตัดสินใจโดยไม่มีทฤษฎีใดๆ มาช่วย โดยกฎที่ได้นี้อาจได้จากการใช้ผู้เชี่ยวชาญ หรือได้จากการใช้วิธีลองผิดลองถูก

ฮิวริสติกส์มักจะถูกใช้ในการแก้ปัญหาที่มีโครงสร้างไม่ชัดเจน และสามารถใช้ในการหาทางเลือกแก้ปัญหาที่น่าพึงพอใจสำหรับปัญหาที่ซับซ้อนหรือปัญหาที่มีโครงสร้าง ได้เร็วกว่าและถูกกว่าการใช้อัลกอริธึม ปัญหาที่เกิดขึ้นในการใช้ ฮิวริสติกส์ก็คือ เป็นวิธีการที่ไม่เป็นที่รู้จักอย่างแพร่หลายเท่ากับการใช้อัลกอริธึม ดังนั้นจึงสามารถนำไปใช้กับสถานการณ์เฉพาะบางอย่างที่มีการวางเป้าหมายเอาไว้ตั้งแต่ต้นเท่านั้น อีกปัญหาหนึ่งก็คือทางแก้ปัญหาที่ได้จากการใช้ฮิวริสติกส์อาจไม่ใช่ทางแก้ปัญหาคือที่สุดเหมือนการใช้อัลกอริธึมอื่น แต่เป็นเพียงทางแก้ปัญหาก็เป็นไปได้และ "ดีเพียงพอ" โดย คำว่า "ดีเพียงพอ" มักจะอยู่ในช่วง 90-99.9% ของทางแก้ปัญหาก็เหมาะสมจริงๆ

### 2.4.1 วิธีการของฮิวริสติกส์

การคิดแบบฮิวริสติกส์ไม่จำเป็นต้องดำเนินไปตามแนวทางการจัดการที่วางไว้ มันจะเกี่ยวข้องกับการค้นหา การเรียนรู้ การประเมินค่า และการตัดสินใจ โดยขบวนการในการค้นหา การเรียนรู้ และการประเมินค่านี้จะเกิดขึ้นซ้ำแล้ว ซ้ำเล่า เหมือนกับการสำรวจ เพื่อนำไปสู่วิธีการอีกรูปแบบหนึ่ง ความรู้จะถูกได้รับจากความสำเร็จหรือความล้มเหลวที่บางจุด ที่มีผลสะท้อน (feed back) กลับมา และทำการแก้ไขขบวนการค้นหานี้ๆ ให้ดีขึ้น

### เมื่อไรที่ควรใช้ฮิวริสติกส์

ต่อไปนี้เป็นสถานการณ์ซึ่งเหมาะสมในการนำฮิวริสติกส์มาใช้

- 1) ข้อมูลเข้าไม่แน่นอน หรือมีจำกัด
- 2) ระบบจริงๆ มีความซับซ้อนมากจนกระทั่งไม่สามารถใช้วิธีการหาคำตอบที่เหมาะสมที่สุด (Optimization) ได้
- 3) ไม่มีวิธีการหรืออัลกอริทึม ที่น่าเชื่อถือที่สามารถใช้ได้อย่างแท้จริง
- 4) ใช้เวลาในการคำนวณหาทางแก้ปัญหาที่เหมาะสมที่สุดมากเกินไป
- 5) มีความเป็นไปได้ที่จะปรับปรุงประสิทธิภาพของขบวนการในการหาทางแก้ที่เหมาะสมที่สุด (เช่น โดยการหาจุดเริ่มต้นที่ดีในการแก้ปัญหาที่มีการใช้ ฮิวริสติกส์) แล้วรวมการใช้ฮิวริสติกส์นี้เข้ากับวิธีการหาคำตอบที่เหมาะสมที่สุด
- 6) เป็นปัญหาที่ซับซ้อน ไม่คุ้มค่ากับการใช้วิธีหาคำตอบที่เหมาะสมที่สุด หรือเป็นปัญหาที่ต้องใช้เวลามาก
- 7) เมื่อมีการเกี่ยวข้องกับขบวนการทางสัญลักษณ์ (symbolic) มากกว่าทางตัวเลข
- 8) เมื่อต้องการทำการตัดสินใจอย่างรวดเร็ว โดยไม่สามารถใช้คอมพิวเตอร์เข้ามาช่วยได้ (การใช้ฮิวริสติกส์บางครั้ง ไม่จำเป็นต้องใช้คอมพิวเตอร์เข้ามาช่วย)

#### 2.4.2 ข้อดีของฮิวริสติกส์

- 1) ง่ายในการทำความเข้าใจ และง่ายในการนำไปใช้และการอธิบาย
- 2) ช่วยในการอบรมคนให้มีความคิดสร้างสรรค์และก่อให้เกิดการสร้างฮิวริสติกส์กับปัญหาอื่นๆ ได้ด้วย
- 3) ประหยัดเวลาในการสร้างตัวแบบ
- 4) ลดความต้องการในด้านการเขียน โปรแกรมและความต้องการด้านแหล่งเก็บข้อมูลคอมพิวเตอร์ได้
- 5) ลดเวลาในการทำงานของคอมพิวเตอร์ จึงทำการตัดสินใจได้อย่างรวดเร็ว
- 6) ทำให้เกิดทางแก้ปัญหาได้หลายทาง
- 7) สามารถประยุกต์ใช้ฮิวริสติกส์ที่มีประสิทธิภาพ เข้ากับตัวแบบซึ่งสามารถแก้ปัญหาด้านการโปรแกรมเชิงคณิตศาสตร์ได้

### 2.4.3 ปัญหาในการใช้วิธีสถิติ

- 1) ไม่สามารถรับประกันได้ว่าทางแก้ปัญหานั้นได้จะเป็นทางแก้ปัญหาคือดีที่สุดหรือเหมาะสมที่สุด
- 2) มีข้อบกพร่องมากเกินไป ในการสร้างเป็นกฎขึ้น
- 3) ผลลัพธ์ที่ได้จากทางเลือกในการตัดสินใจ อาจไม่เป็นไปตามที่คาดเอาไว้
- 4) การขึ้นระหว่างกันของส่วนใดส่วนหนึ่งของระบบ บางครั้งสามารถมีอิทธิพลอย่างมากกับระบบทั้งระบบได้

นักวิจัยบอกว่าการใช้วิธีต่างๆ ไปและการใช้วิธีสถิติสามารถจะเกิดการผิดพลาดได้ เพราะวิธีเหล่านั้น ไม่มี ข้อกำหนดตายตัว คือ ไม่มีข้อจำกัดว่าจะต้องเริ่มต้นที่จุดไหน, ลำดับขั้นตอนการทำงานต้องเป็นอย่างไร หรือ ไม่จำกัดว่าจะต้องสร้างตัวเลือกในการตัดสินใจหรือไม่, ไม่เจาะจงด้านข้อจำกัดของการแก้ปัญหาลักษณะของเกณฑ์ที่ใช้ในการระบุขบวนการทำงาน ระดับของค่าใช้จ่ายที่ใช้ในการหาว่าผลลัพธ์สุดท้ายที่ได้เป็นผลลัพธ์ที่ดีที่สุดจริงๆ ผลลัพธ์เป็นพฤติกรรมที่ไม่มีจุดมุ่งหมายแน่ชัดและไม่สามารถคาดเดาได้ ผลลัพธ์อาจดีในการนำไปใช้กับระบบงานหนึ่งแต่อาจไม่ดีในการนำไปใช้กับอีกระบบงานอื่นก็ได้

### 2.4.4 การจำลองเหตุการณ์

การจำลองเหตุการณ์ เป็นการดำเนินการบนตัวแบบที่ถูกสร้างขึ้นเพื่อเป็นตัวแทนระบบจริงในช่วงหนึ่งๆ เพื่อศึกษาพฤติกรรมของระบบ และนำผลที่ได้จากตัวแบบไปใช้ในการพยากรณ์พฤติกรรมของระบบงานที่แท้จริง ในการจำลองเหตุการณ์จะใช้ข้อมูลเก่าหรือข้อมูลที่วางแผนไว้ก็ได้ โดยตัวแบบที่ใช้สามารถแสดงผลการคำนวณต่างๆ ได้ เช่น กำไรรวม (หรือ ขาดทุนรวม) เปอร์เซ็นต์ของลูกค้าที่ไม่พอใจ และจำนวนสินค้าที่เหลือ เป็นต้น

คุณลักษณะหลักของการจำลองเหตุการณ์

- 1) เป็นการจำลองระบบในความเป็นจริง ซึ่งมีการทำงานกับความเป็นจริงที่ง่ายกว่าแบบจำลองอื่นๆ
- 2) เป็นวิธีการสำหรับนำไปสู่การทดลอง ดังนั้นการจำลองเหตุการณ์จะเกี่ยวข้องกับการทดสอบค่าที่กำหนดไว้ ในการตัดสินใจหรือตัวแปรที่ไม่สามารถควบคุมได้ในแบบจำลอง และทำการสังเกตผลที่เกิดขึ้นตามมา

3) เป็นเครื่องมือแบบเชิงบรรยาย จึงสามารถอธิบาย และ/หรือคาดเดาค่าคุณลักษณะของระบบที่กำหนดภายใต้สถานการณ์ต่างๆ ได้

4) การจำลองเหตุการณ์มักจะประกอบด้วยการทำงานทดลองหลายๆ ครั้งเพื่อประมาณผลกระทบที่จะเกิดจากการกระทำนั้นๆ

5) มักจะใช้กับปัญหาที่มีความยุ่งยากเกินกว่าที่จะใช้วิธีการหาคำตอบที่เหมาะสมที่สุดเช่น การโปรแกรมเชิงเส้นได้

ข้อดีของการจำลองเหตุการณ์

- 1) ลดเวลาในการทำงาน (ถ้ามีการนำคอมพิวเตอร์เข้ามาใช้)
- 2) เป็นทฤษฎีที่มีการเปรียบเทียบอย่างชัดเจน
- 3) ง่ายในการรวบรวมส่วนต่างๆ ที่สัมพันธ์กันและขึ้นต่อกัน
- 4) เป็นแบบจำลองเชิงบรรยาย (descriptive model) ซึ่งทำให้ผู้บริหารสามารถถามคำถามประเภทเงื่อนไข (what-if) ได้ ดังนั้นวิธีนี้จะเร็วและถูกกว่า และเสี่ยงน้อยกว่าวิธีลองผิดลองถูก
- 5) ตัวแบบที่มีความแม่นยำต้องมีความรู้เกี่ยวกับปัญหานั้นๆ ดังนั้นจึงเป็นการบังคับให้ผู้สร้างระบบสนับสนุนการบริหารมีการประสานงานกับผู้บริหารอย่างดี
- 6) เป็นแบบจำลองที่สร้างจากมุมมองของผู้บริหารและสร้างตามโครงสร้างการตัดสินใจของผู้บริหารคนนั้นๆ
- 7) ตัวแบบในการจำลองเหตุการณ์หนึ่งๆ ถูกสร้างขึ้นเพื่อใช้กับปัญหาเฉพาะ ไม่เกี่ยวกับปัญหาอื่นๆ ดังนั้น ผู้จัดการไม่จำเป็นต้องรู้และเข้าใจในทุกๆ ส่วนที่อยู่ภายในแบบจำลองก็ได้
- 8) การจำลองเหตุการณ์ สามารถจัดการกับปัญหาได้หลายประเภท
- 9) ผู้บริหารสามารถทดลองกับตัวแปรต่างๆ เพื่อพิจารณาเลือกทางเลือกที่ดีที่สุดในกลุ่มได้

10) อนุญาตให้มีการรวมความซับซ้อนของปัญหาจริงๆ เข้าไว้ด้วยก็ได้ เช่นมีการนำการกระจายความน่าจะเป็นที่เกิดขึ้นจริงๆ เข้ามาใช้

11) ง่ายในการหาประสิทธิภาพต่างๆ

ข้อจำกัดของการจำลองเหตุการณ์

- 1) แต่ละทางเลือกต้องถูกทดลองอย่างสมบูรณ์
- 2) ไม่สามารถรับประกันได้ว่าการแก้ปัญหานั้นเหมาะสมที่สุดจริงๆ อาจเป็นเพียงทางเลือกที่ดีที่สุดในงานที่ทำการทำงานทดลองเท่านั้น
- 3) ต้องการผู้เชี่ยวชาญมาช่วยในการออกแบบการจำลองเหตุการณ์ และการ โปรแกรมลงไปในคอมพิวเตอร์ และแปลผลลัพธ์ทางสถิติ



- 4) ค่าใช้จ่ายที่ใช้ในการสร้างและทดสอบอาจค่อนข้างสูง
- 5) การสร้างตัวแบบในการจำลองเหตุการณ์มักจะใช้เวลานาน
- 6) การแก้ปัญหาโดยการศึกษจากการจำลองเหตุการณ์ มักจะไม่สามารถนำไปใช้กับปัญหาอื่น ๆ ได้ (เพราะมีการใช้ปัจจัยเฉพาะกับปัญหานั้นๆ เลย)

#### 2.4.5 ประเภทของการจำลองเหตุการณ์

การจำลองเหตุการณ์ที่เป็นที่นิยมมากได้แก่

- 1) การจำลองเหตุการณ์เชิงความน่าจะเป็น (Probabilistic Simulation) ในการจำลองเหตุการณ์แบบนี้ ตัวแปรอิสระมีความน่าจะเป็นเข้ามาเกี่ยวข้องด้วย สามารถแบ่งประเภทย่อยออกเป็น
  - 2) การจำลองเหตุการณ์ที่มีการกระจายแบบไม่ต่อเนื่อง (Discrete distributions) เป็นการจำลองเหตุการณ์ที่เกี่ยวข้องกับสถานการณ์ที่มีเหตุการณ์(หรือตัวแปร)จำกัด
  - 3) การจำลองเหตุการณ์ที่มีการกระจายแบบต่อเนื่อง (Continuous distributions) เป็นการจำลองเหตุการณ์ที่เกี่ยวข้องกับสถานการณ์ที่มีเหตุการณ์ที่เป็นไปได้เป็นจำนวนไม่จำกัด
 นอกจากนี้ยังสามารถแบ่งประเภทการจำลองเหตุการณ์ตามการขึ้นต่อเวลาได้เป็น
  - 1) การจำลองเหตุการณ์ที่ขึ้นต่อช่วงเวลา (Time Dependent) หมายถึงสถานการณ์ที่ต้องทราบเวลาที่แน่นอนในการเกิดเหตุการณ์
  - 2) การจำลองเหตุการณ์ที่เป็นอิสระต่อช่วงเวลา (Time independent) หมายถึงสถานการณ์ที่ไม่จำเป็นต้องทราบว่าเมื่อไรที่เหตุการณ์นั้นๆ เกิดขึ้น

#### 2.5 การจัดลำดับงานของเครน

ปัจจุบันการขนส่งทางเรือเป็นการขนส่งที่สำคัญและมีมากขึ้นเรื่อยๆ ดังนั้นจึงต้องการระบบจัดการที่มีประสิทธิภาพ ดังเช่น ท่าเรือสิงคโปร์ (The Port of Singapore Authority; PSA) เป็นหนึ่งในท่าเรือที่มีปริมาณการขนส่งและลำเลียงคอนเทนเนอร์ที่มากที่สุดในโลก ซึ่ง มีคอนเทนเนอร์ที่ต้องจัดการถึง 17.04 ล้าน TEU (Twenty-foot equivalent unit) โดยคิดเป็น 9% ของคอนเทนเนอร์ทั้งหมดของโลก ขณะนี้ PSA ต้องการที่จะเพิ่มขีดความสามารถในการจัดการกับคอนเทนเนอร์เหล่านี้ แต่มีปัญหาที่มีขนาดของท่าเรือจำกัด เครื่องจักรมีไม่เพียงพอต่อการลำเลียง

การจัดลำดับการทำงานของเครนมีความสำคัญมากในการจัดระบบของท่าเรือ เครนจะทำหน้าที่เชื่อมต่อระหว่างทะเลกับพื้นดิน เพื่อลำเลียงคอนเทนเนอร์ลงจากเรือ จากการวิจัยพบว่าการทำงานของเครนเป็นคอขวดของกระบวนการปฏิบัติงานของท่าเรือ ในขณะเดียวกัน Kon และคณะ (1994) พบว่าปัญหาในการตัดสินใจส่วนมากใช้ประสบการณ์และการจำลองสถานการณ์ (Simulation) เป็นเครื่องมือในขณะที่การจำลองสถานการณ์จะเป็นเครื่องมือที่มีประโยชน์อย่างยิ่ง

แต่วิธีเชิงวิเคราะห์ (Analytical Method) ซึ่งให้นักคณิตศาสตร์เข้ามาช่วยนั้น ก็จะเพิ่มความสามารถในการตัดสินใจได้ดียิ่งขึ้น

ระบบการทำงานของเครนของท่าเรือค่อนข้างจะแตกต่างจากระบบในอุตสาหกรรมอื่นๆ ดังนั้นจึงมีแบบจำลองที่ถูกพัฒนาขึ้นมาโดย Lim และคณะ (2003) ได้เสนอการศึกษาการจัดลำดับงานของเครน โดยมีข้อจำกัดทางด้านพื้นที่ ซึ่งข้อจำกัดอย่างหนึ่งที่สำคัญคือการห้ามข้ามตัดกันของเครน (Non-Crossing Constraint) กล่าวคือ แขนของเครนไม่สามารถที่จะข้ามกันได้ในเวลาหนึ่งๆ ในงานวิจัยนี้เราจะศึกษาการจัดลำดับของเครน ที่ได้พัฒนาขึ้นและการประยุกต์โดยใช้วิธี ฮิวริสติกส์เพื่อลดเวลาของการประมวลผล จากโปรแกรมสำเร็จรูป



1506 014X

ร.ร.

01986

2552

## บทที่ 3

### วิธีดำเนินโครงการ

#### 3.1 ศึกษาการใช้วิธีการแบบฮิวริสติกส์เพื่อมาประยุกต์ในการหาการทำงานของแกรม

ในขั้นตอนนี้เราจะศึกษาการคิดและทำงานแบบฮิวริสติกส์ โดยการทดลองออกแบบการคิดคำนวณ วิธีฮิวริสติกส์นั้นเป็นการใช้สัญชาตญาณในการหาและวิเคราะห์โดยใช้ประสบการณ์ ดังนั้นจึงต้องออกแบบหลักการทำให้เป็นไปในรูปแบบธรรมชาติมากที่สุด ซึ่งจะส่งผลให้ได้ผลลัพธ์ของเวลาในการประมวลผลที่รวดเร็ว และเป็นคำตอบที่สามารถนำไปใช้ได้ โดยเน้นการหาคำตอบในเวลาอันสั้นแม้จะมีข้อมูลที่ต้องการหาขนาดใหญ่ก็ตาม

#### 3.2 ศึกษาโปรแกรม Python

ขั้นตอนการศึกษาโปรแกรม Python นั้น ต้องมาทำความเข้าใจกันก่อนว่าโปรแกรม Python เป็นโปรแกรมที่สามารถนำไปใช้ได้หลากหลายรูปแบบมาก แต่ที่เรานำมาใช้เป็นการนำโปรแกรมนี้มาเขียนให้เป็นในรูปแบบของโปรแกรมคำนวณ โดยใช้วิธีการแบบฮิวริสติกส์เป็นแนวทางในการเขียนโปรแกรม

การศึกษการเขียนโปรแกรม Python นี้ จะแบ่งออกเป็นศึกษาโครงสร้างและการใช้งานโปรแกรมแบบทั่วไป การออกแบบและเขียนโปรแกรมในรูปแบบที่กำหนด การใช้โปรแกรมเพื่อการคำนวณหาผลลัพธ์ที่ต้องการ และการสรุปผลของโปรแกรม

#### 3.3 เขียนโปรแกรม Python โดยการใช้วิธีการแบบฮิวริสติกส์มาประยุกต์

การเขียนโปรแกรม Python นั้นโดยทั่วไปแล้วสามารถเขียนโปรแกรมนี้ให้ทำงานแบบใดก็ได้ แต่ทางผู้วิจัยได้กำหนดหลักการทำงานไว้ว่าจะให้โปรแกรม Python ทำงานออกมาในรูปแบบของวิธีฮิวริสติกส์ โดยมีจุดประสงค์เพื่อใช้เวลาในการหาคำตอบที่ไวที่สุด การเขียนโปรแกรมโดยวิธีนี้จะมีใจความสำคัญคือการออกแบบให้โปรแกรมที่เขียนนั้นมีความใกล้เคียงกับการทำงานของหลักการคิดของมนุษย์มากที่สุด ซึ่งก็เป็นหลักการของวิธีฮิวริสติกส์นั่นเอง ซึ่งผลลัพธ์ของวิธีการทำงานแบบนี้เองจะมีผลลัพธ์ของเวลาการคำนวณที่เร็วมากอีกทั้งยังให้ค่าของการคำนวณที่สามารถนำไปใช้ได้จริงอีกด้วย

### 3.4 ศึกษาการโปรแกรมสำเร็จรูป

โปรแกรมสำเร็จรูปนี้ เป็นโปรแกรมที่สามารถคำนวณผลลัพธ์ในรูปแบบของสมการเชิงคณิตศาสตร์ที่ให้ผลลัพธ์ที่ดีที่สุด แต่จะใช้เวลาในการประมวลผลที่นานซึ่งขึ้นอยู่กับขนาดของปัญหาของงานด้วย โดยที่เราจะศึกษาเกี่ยวกับโปรแกรมนี้คือ หลักการทำงานทั่ว ๆ ไปของโปรแกรมและการเขียน โปรแกรมจากสมการที่มีตลอดจนการทดสอบหาค่าผลลัพธ์และเวลาของการคำนวณ

### 3.5 ทำการทดลองเปรียบเทียบเวลาการประมวลผลระหว่างโปรแกรม Python ที่เขียนขึ้นกับโปรแกรมสำเร็จรูป

ในขั้นตอนนี้จะทำการเปรียบเทียบผลลัพธ์ของการทำงานของทั้งสอง โปรแกรม ทั้งในเรื่องของค่าผลลัพธ์ที่ได้และเวลาที่ใช้ในการประมวลผลเพื่อนำมาจัดทำและสรุปผลถึงความแตกต่างของการทำงานของทั้ง 2 โปรแกรม

### 3.6 สรุปผลและจัดทำรายงาน

นำข้อมูลที่ทำการศึกษาได้จากการประมวลผล โดยโปรแกรมจากสมการทางคณิตศาสตร์ของทั้ง 2 รูปแบบ มาสรุปจัดทำเป็นรูปเล่มเพื่อเป็นเอกสารในการนำเสนอและเป็นแหล่งอ้างอิงจากการวิจัยเรื่องการจัดลำดับการทำงานของครน โดยใช้โปรแกรมเชิงคณิตศาสตร์

### 3.7 นำเสนองานวิจัย

นำงานที่แล้วเสร็จเป็นรูปเล่มมาทำการนำเสนอในรูปแบบของงานวิจัยเรื่องการจัดลำดับการทำงานของครน โดยใช้โปรแกรม Python ประมวลผลเพื่อแสดงคำตอบ เวลาในการประมวลผลและแสดงสมการทางคณิตศาสตร์ที่ได้ในการใช้วิธีฮิวริสติกส์มาประยุกต์

## บทที่ 4

### ผลการทดลองและวิเคราะห์

ในบทนี้ทางผู้วิจัยจะอธิบายหลักการการทำงานของแต่ละ โปรแกรมที่สร้างขึ้นเพื่อแก้ปัญหาการจัดลำดับการทำงานของครน โดยสรุปในรูปแบบของตารางแสดงผลต่างๆ ที่สามารถอ่านค่าและทำความเข้าใจได้ง่ายเพื่อนำไปวิเคราะห์และสรุปผลการทดลองในบทต่อไป

#### 4.1 วิธีวิธีวิสตติคส์ที่ได้พัฒนาขึ้นเพื่อแก้ปัญหาบนโปรแกรม Python

โปรแกรม Python ที่เขียนด้วยโปรแกรมที่ใช้วิธีวิสตติคส์ในการออกแบบนั้น ทางผู้วิจัยได้จัดทำขึ้น โดยการนำโปรแกรมทั้ง 7 โปรแกรมเข้ารวมกันเพื่อประมวลผลคำตอบที่เหมาะสมบนโปรแกรม Python ในที่นี้ทางผู้วิจัยจะตั้งชื่อว่า Program Python for quick crane scheduling หรือเรียกย่อว่า PQCS เพื่อง่ายต่อการอธิบายและเพื่อการเรียกชื่อได้เข้าใจตรงกัน ในหัวข้อนี้เราจะแสดงโปรแกรมต่างๆที่สร้างขึ้น โดยแยกแต่ละ โปรแกรมออกมาเพื่ออธิบายหลักการการทำงานซึ่งโปรแกรมที่สร้างขึ้นแบ่งเป็นหัวข้อย่อยได้ดังนี้

##### 4.1.1 วิธี Greedy 1

โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่าภาระการทำงานของครนนั้นสามารถหาค่าได้โดย การปล่อยให้ครนที่ 1 ทำงานตามลำดับด้วยปัญหาที่สร้างขึ้น และเมื่อก่อนถึงค่าเฉลี่ยของปัญหาทั้งหมดให้ทำการตัดการทำงานเพื่อเริ่มการทำงานใหม่ที่ครนลำดับต่อไป โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

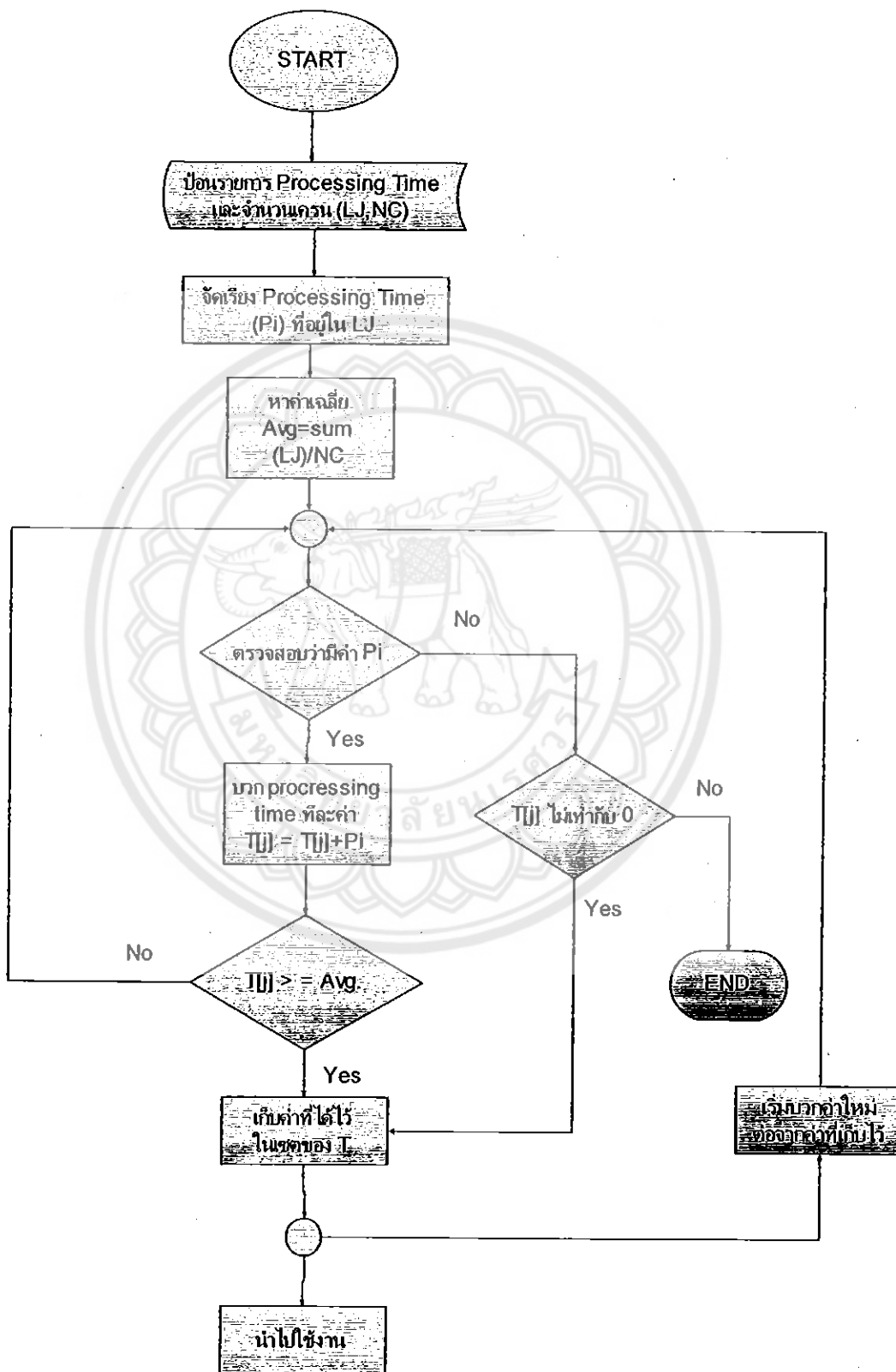
ขั้นตอนการทำงานของ โปรแกรม Greedy 1

- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนครน)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อครน ( $Avg = \sum X / K$ ) เมื่อ X คือ Processing Time และ K คือ จำนวนครน
- 4) นับจำนวนของ Processing Time สะสมจากซ้ายไปขวา เมื่อจำนวนสะสมเกินค่าเฉลี่ยทำใด ให้ตัดที่จำนวนนั้น

- 5) เริ่มนับจำนวนสะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนงานหมด

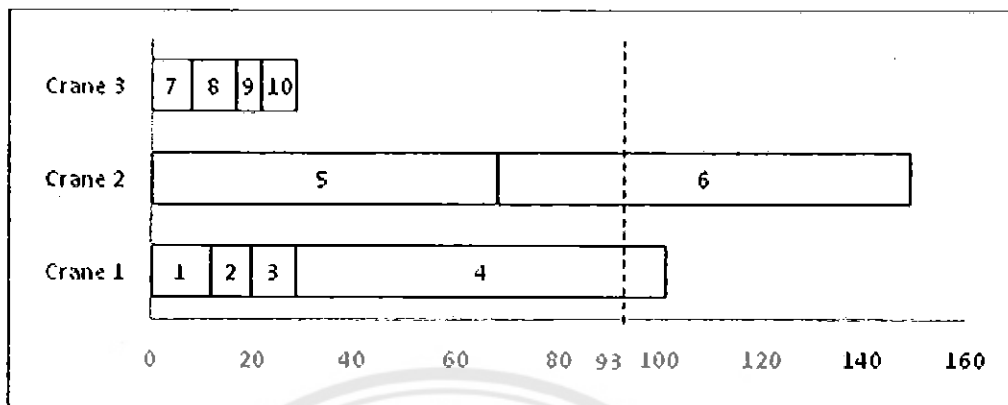


## แผนผังการทำงานของ โปรแกรม Greedy 1



รูปที่ 4.1 แผนผังการทำงานของ โปรแกรม Greedy 1

### แผนผังตัวอย่างการทำงานในแต่ละเครนของโปรแกรม Greedy 1



#### รูปที่ 4.2 แสดงการทำงานในแต่ละเครนของ Greedy 1 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.2 อธิบายการคำนวณตามวิธี Greedy 1 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลางมีข้อมูลของ Processing Time คือ 12,8,9,72,68, 81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 1 คือการคำนวณให้แต่ละเครนมีการทำงานที่เกินค่าเฉลี่ยมีวิธีการทำงานคือ เครนที่ 1 เลือกทำงานที่ 1 จากนั้นทำการบวกรงานที่ 2 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรงานที่ 3 เพิ่ม พบว่ายังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรงานที่ 4 เข้าไป ตรวจผลลัพธ์ปรากฏว่ามากกว่าค่าเฉลี่ย เริ่มการคำนวณค่าของเครนที่ 2 โดยเริ่มที่งานที่ 5 จากนั้นทำการบวกรงานที่ 6 เพิ่ม ปรากฏว่าผลลัพธ์เกินค่าเฉลี่ย จึงทำการเริ่มคำนวณค่าของเครนที่ 3 โดยนำงานที่เหลือมารวมกันโดยมีค่าเวลาการทำงานสูงที่สุดที่เครน 2 ใช้เวลา 149 วินาที

#### 4.1.2 วิธี Greedy 2

โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่าเราจะหาการทำงานของเครนนั้นสามารถหาค่าได้โดย การปล่อยให้เครนที่ 1 ทำงานตามลำดับด้วยค่าสะสมของปัญหาจากลำดับสุดท้ายของปัญหามาจนถึงปัญหาแรก และเมื่อก่อนถึงค่าเฉลี่ยของปัญหาทั้งหมดให้ทำการตัดการทำงานเพื่อเริ่มการทำงานใหม่ที่เครนลำดับต่อไปโดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

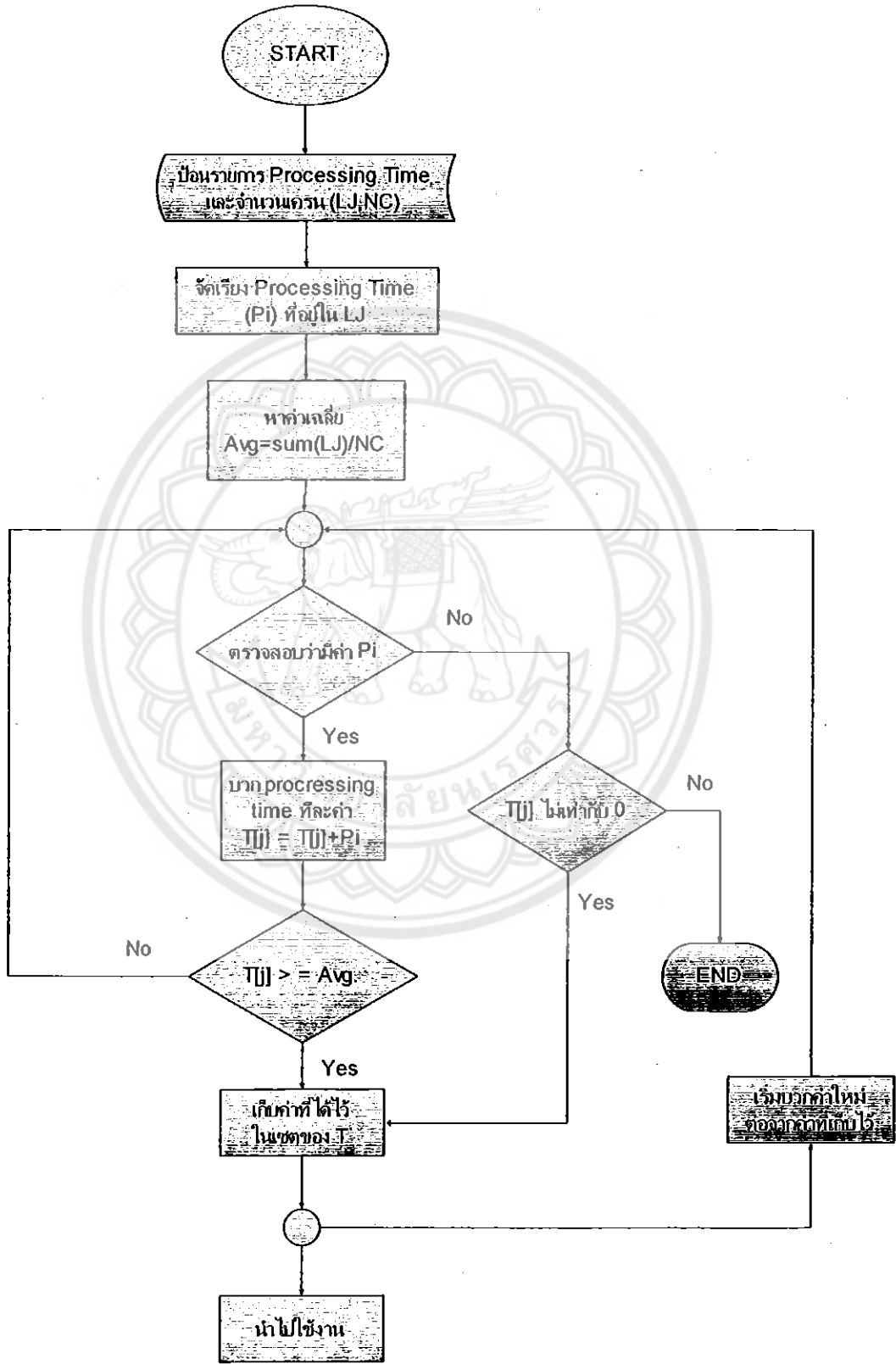


ขั้นตอนการทำงานของ โปรแกรม Greedy 2

- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนเครน)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อเครน ( $Avg = \sum X / K$ ) เมื่อ X คือ Processing Time และ K คือ จำนวนเครน
- 4) นับจำนวนของ Processing Time สะสมจากขวามาซ้าย เมื่อจำนวนสะสมเกินค่าเฉลี่ยที่ค่าใด ให้ตัดที่จำนวนนั้น
- 5) เริ่มนับจำนวนสะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนงานหมด

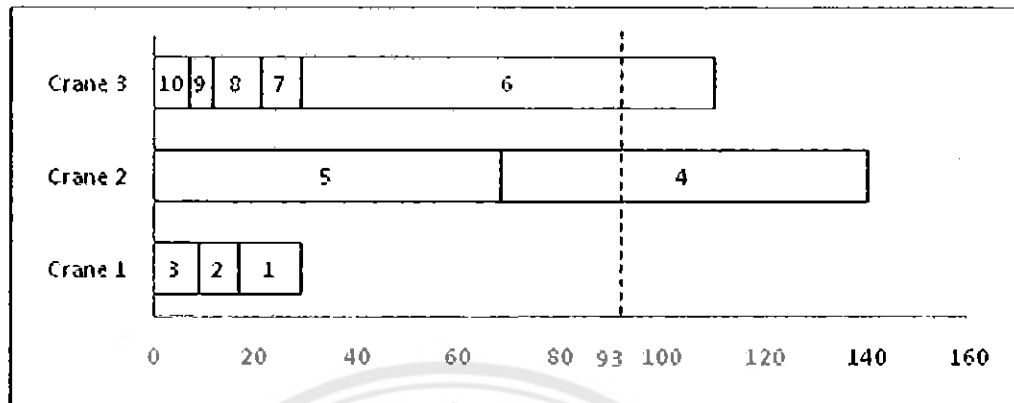


แผนผังการทำงานของโปรแกรม Greedy 2



รูปที่ 4.3 แผนผังการทำงานของโปรแกรม Greedy 2

### แผนผังตัวอย่างการทำงานในแต่ละครนของโปรแกรม Greedy 2



#### รูปที่ 4.4 แสดงการทำงานในแต่ละครนของ Greedy 2 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.4 อธิบายการคำนวณตามวิธี Greedy 2 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68,81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 2 คือการคำนวณให้แต่ละครนมีการทำงานที่เกินค่าเฉลี่ยและทำการหาค่าจากงานลำดับสุดท้ายมาลำดับงานแรก โดยมีวิธีการทำงานคือ ครนที่ 3 เลือกทำงานที่ 10 จากนั้นทำการบวกรงานที่ 9 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรงานที่ 8 เพิ่มพบว่ายังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรงานที่ 7 เข้าไปพบว่ายังไม่ถึงค่าเฉลี่ยจึงทำการบวกรงานที่ 6 เข้าไป ตรวจผลลัพธ์ปรากฏว่ามากกว่าค่าเฉลี่ย เริ่มการคำนวณค่าของครนที่ 2 โดยเริ่มที่งานที่ 5 พบว่ายังไม่ถึงค่าเฉลี่ย จากนั้นทำการบวกรงานที่ 4 เพิ่ม ปรากฏว่าผลลัพธ์เกินค่าเฉลี่ย จึงทำการเริ่มคำนวณค่าของครนที่ 1 โดยนำงานที่เหลือมารวมกัน โดยมีค่าเวลาการทำงานสูงสุดที่ครน 2 ใช้เวลา 140 วินาที

#### 4.1.3 วิธี Greedy 3

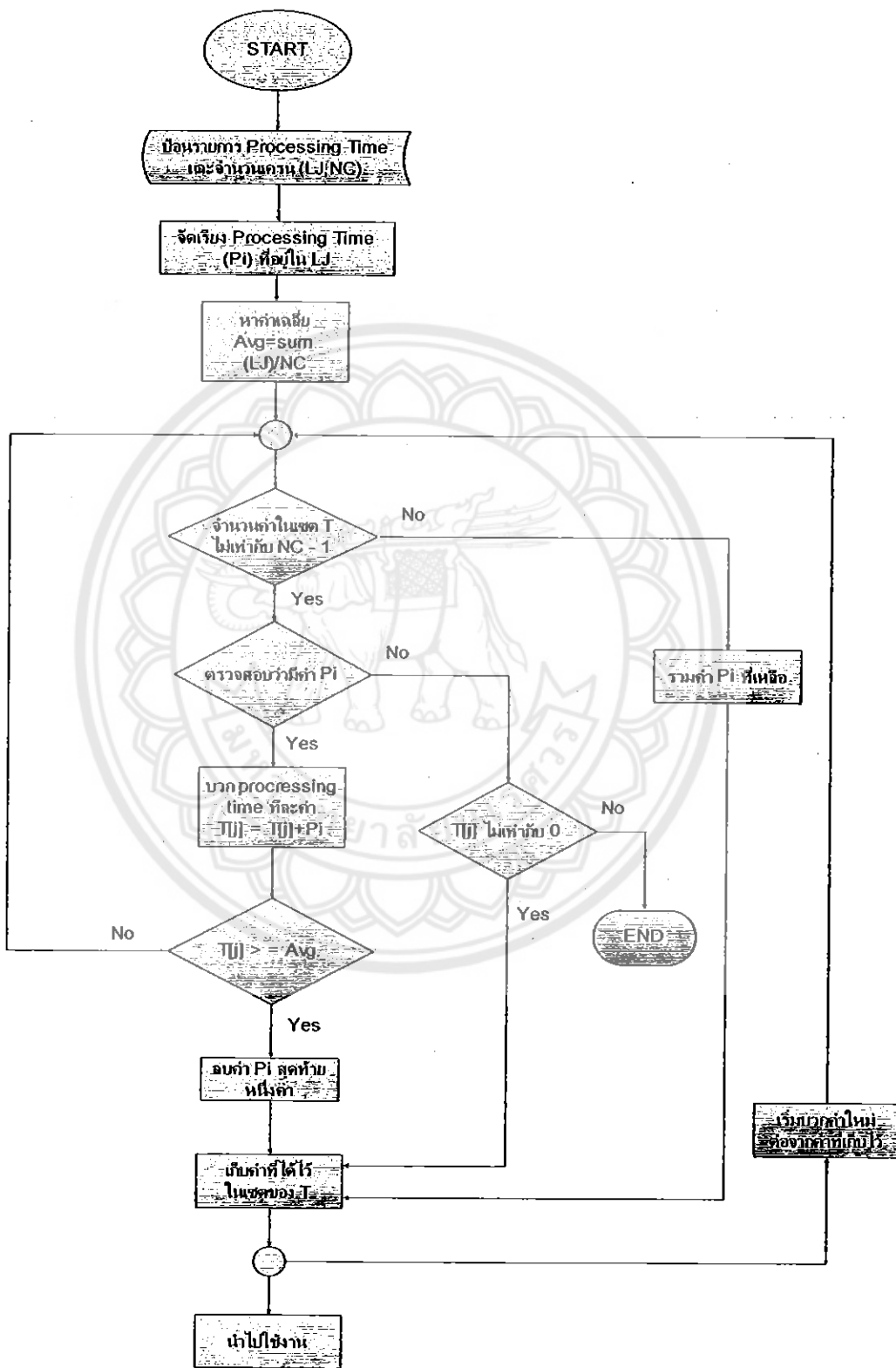
โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่ากระจะหาการทำงานของครนนั้นสามารถหาค่าได้โดย การปล่อยให้ครนที่ 1 ทำงานตามลำดับด้วยปัญหาที่สร้างขึ้น และหยุดเมื่อเกินค่าเฉลี่ยของปัญหาทั้งหมดจึงทำการตัดการทำงานเพื่อเริ่มการทำงานใหม่ที่ครนลำดับต่อไป โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

### ขั้นตอนการทำงานของโปรแกรม Greedy 3

- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนเครื่อง)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อเครื่อง ( $Avg = \sum X / K$ ) เมื่อ X คือ Processing Time และ K คือ จำนวนเครื่อง
- 4) นับจำนวน Processing Time สะสมจากซ้ายไปขวา เมื่อจำนวนสะสมเกินค่าเฉลี่ยที่งานใด ให้ตัดที่งานก่อนหน้านั้น
- 5) เริ่มนับจำนวน Processing Time สะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนได้จำนวนกลุ่มเท่ากับจำนวนเครื่อง - 1 ให้เครื่องตัวสุดท้ายทำงานที่เหลือ

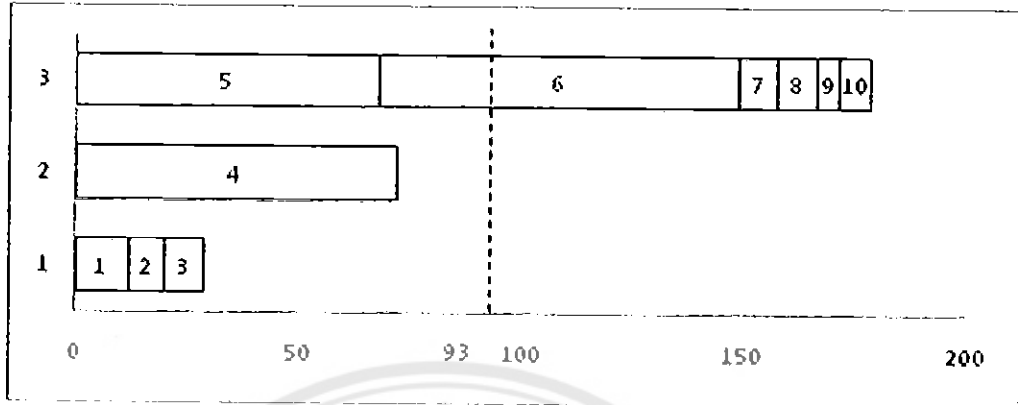


แผนผังการทำงานของโปรแกรม Greedy 3



รูปที่ 4.5 แผนผังการทำงานของโปรแกรม Greedy 3

แผนผังตัวอย่างการทำงานในแต่ละเครื่องของโปรแกรม Greedy 3



รูปที่ 4.6 แสดงการทำงานในแต่ละเครื่องของ Greedy 3 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.6 อธิบายการคำนวณตามวิธี Greedy 3 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68, 81,8,9,5,7 โดยนับตามลำดับ เช่นข้อมูลที่ 1 ใช้เวลาในการทำงาน 12 วินาที ข้อมูลที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 3 คือการคำนวณให้แต่ละเครื่องมีการทำงานที่ต่ำกว่าค่าเฉลี่ยโดยเรียงลำดับข้อมูลจากลำดับแรกไปลำดับสุดท้ายมีวิธีการทำงานคือ เครื่องที่ 1 เลือกทำงานที่ 1 จากนั้นทำการบวกรงานที่ 2 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรงานที่ 3 เพิ่มยังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรงานที่ 4 เข้าไป ตรวจผลลัพธ์ปรากฏว่ามากกว่าค่าเฉลี่ยหยุดการทำงานของเครื่อง 1 ที่งานที่ 3 เริ่มการคำนวณค่าของเครื่องที่ 2 โดยเริ่มที่งานที่ 4 จากนั้นทำการบวกรงานที่ 5 เพิ่ม ปรากฏว่าผลลัพธ์เกินค่าเฉลี่ย หยุดการทำงานของเครื่อง 2 ที่งานที่ 4 จึงทำการเริ่มคำนวณค่าของเครื่องที่ 3 โดยนำงานที่ 5 และงานที่ 6 เหลือมารวมกันโดยมีค่าเวลาการทำงานสูงที่สุดที่เครื่อง 3 ใช้เวลา 178 วินาที

#### 4.1.4 วิธี Greedy 4

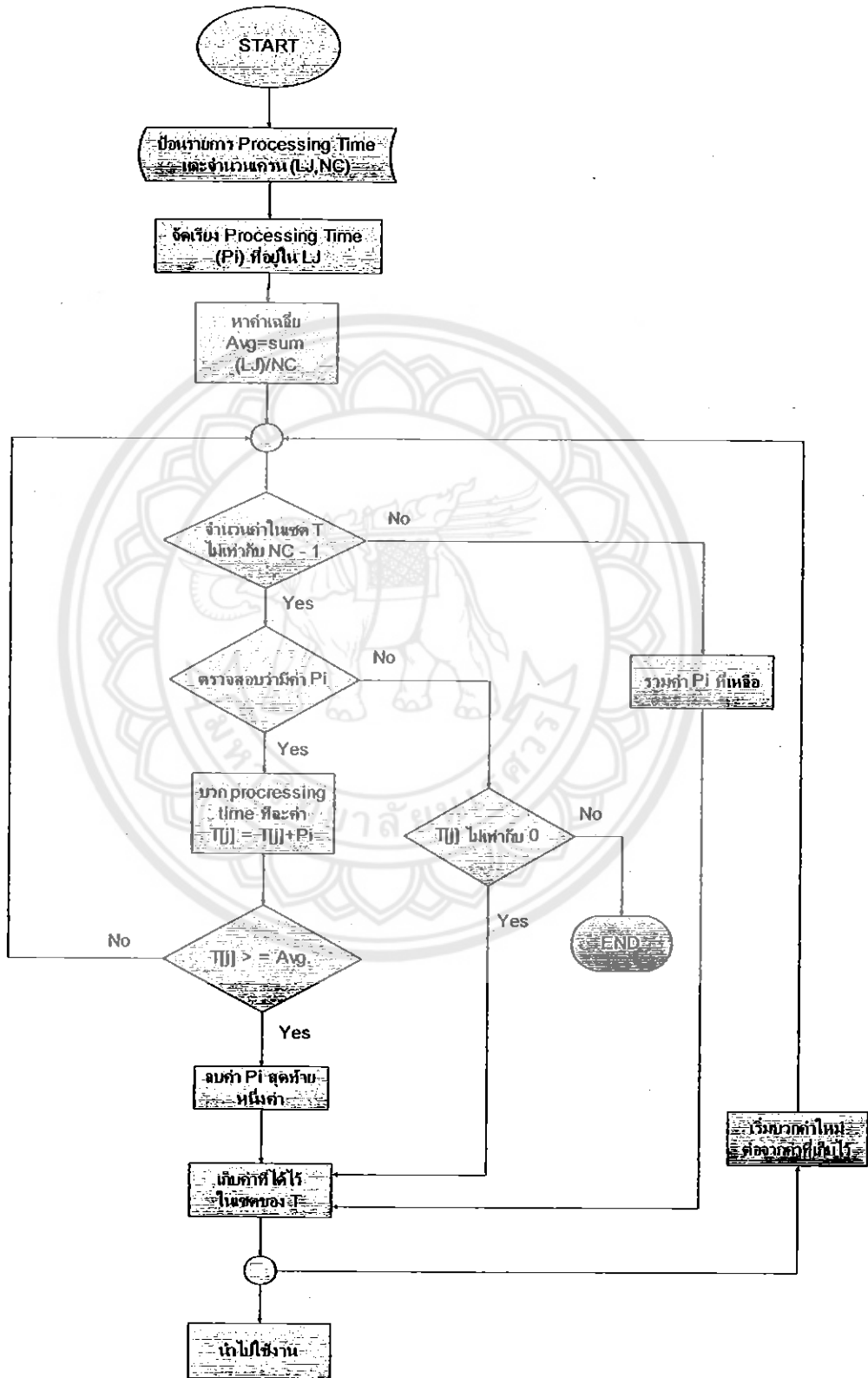
โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่าเราจะหาการทำงานของเครื่องนั้นสามารถหาค่าได้โดย การปล่อยให้เครื่องที่ 1 ทำงานตามลำดับด้วยค่าสะสมของปัญหาจากลำดับสุดท้ายของปัญหามาจนถึงปัญหาแรก และหยุดเมื่อเกินค่าเฉลี่ยของปัญหาทั้งหมดจึงทำการตัดการทำงานเพื่อเริ่มการทำงานใหม่ที่เครื่องลำดับต่อไป โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

#### ขั้นตอนการทำงานของโปรแกรม Greedy 4

- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนเครน)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อเครน ( $Avg = \sum X / K$ ) เมื่อ  $X$  คือ Processing Time และ  $K$  คือ จำนวนเครน
- 4) นับจำนวน Processing Time สะสมจากขวาไปซ้าย เมื่อจำนวนสะสมเกินค่าเฉลี่ยที่งานใด ให้ตัดที่งานก่อนหน้านั้น
- 5) เริ่มนับจำนวน Processing Time สะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนได้จำนวนกลุ่มเท่ากับจำนวนเครน - 1 ให้เครนตัวสุดท้ายทำงานที่เหลือ



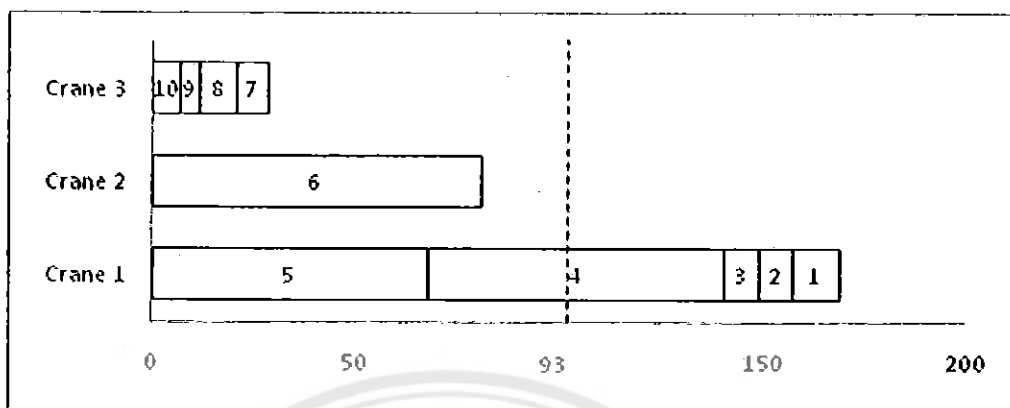
แผนผังการทำงานของโปรแกรม Greedy 4



รูปที่ 4.7 แผนผังการทำงานของโปรแกรม Greedy 4



### แผนผังตัวอย่างการทำงานในแต่ละเครนของโปรแกรม Greedy 4



**รูปที่ 4.8** แสดงการทำงานในแต่ละเครนของ Greedy 4 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.8 อธิบายการคำนวณตามวิธี Greedy 4 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68,81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 4 คือการคำนวณให้แต่ละเครนมีการทำงานที่ต่ำกว่าค่าเฉลี่ยโดยเรียงลำดับข้อมูลจากลำดับสุดท้ายไปลำดับแรกมีวิธีการทำงานคือ เครนที่ 3 เลือกทำงานที่ 10 จากนั้นทำการบวกรวมงานที่ 9 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรวมงานที่ 8 เพิ่มยังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรวมงานที่ 7 เข้าไปยังไม่ถึงค่าเฉลี่ยจึงทำการบวกรวมงานที่ 6 เข้าไป ตรวจผลลัพธ์ปรากฏว่ามากกว่าค่าเฉลี่ยหยุดการทำงานของเครนที่ 3 ที่งานที่ 5 เริ่มการคำนวณค่าของเครนที่ 2 โดยเริ่มที่งานที่ 6 จากนั้นทำการบวกรวมงานที่ 7 เพิ่ม ปรากฏว่าผลลัพธ์เกินค่าเฉลี่ย หยุดการทำงานของเครนที่ 2 ที่งานที่ 6 จึงทำการเริ่มคำนวณค่าของเครนที่ 3 โดยนำงานที่เหลือมารวมกันโดยมีค่าเวลาการทำงานสูงสุดที่เครน 1 ใช้เวลา 169 วินาที

#### 4.1.5 วิธี Greedy 5

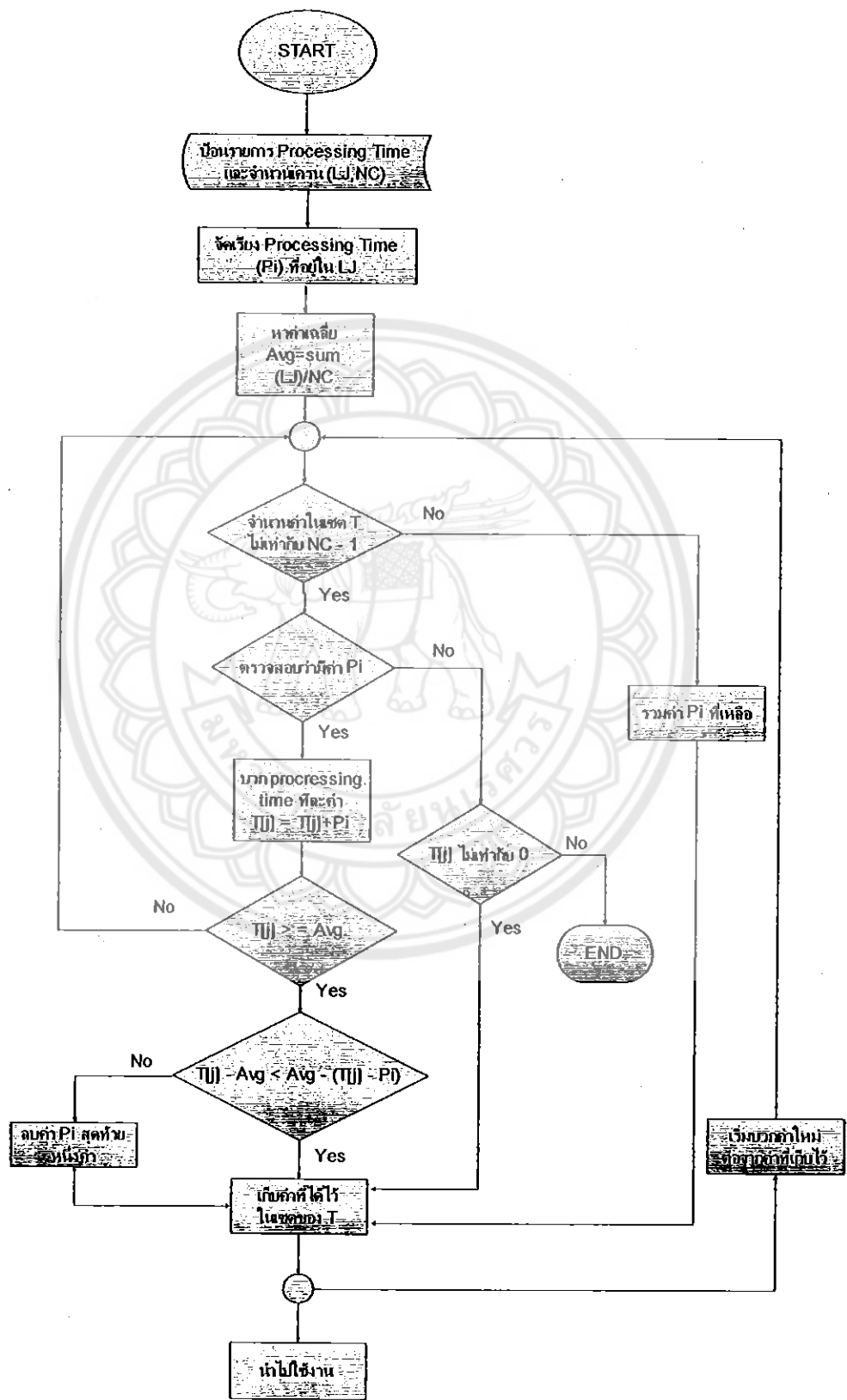
โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่าเราจะหาการทำงานของเครนนั้นสามารถหาค่าได้โดย การปล่อยให้เครนที่ 1 ทำงานตามลำดับด้วยปัญหาที่สร้างขึ้น และเก็บค่าของปัญหาเมื่อเกินค่าเฉลี่ยของปัญหาทั้งหมดและนำมาเปรียบเทียบกับค่าของปัญหาก่อนถึงค่าเฉลี่ยแล้วจึงทำการเลือก

การทำงานที่ใกล้ค่าเฉลี่ยมากที่สุด ก่อนจะเริ่มทำงานที่เครื่องลำดับถัดไป โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

ขั้นตอนการทำงานของโปรแกรม Greedy 5

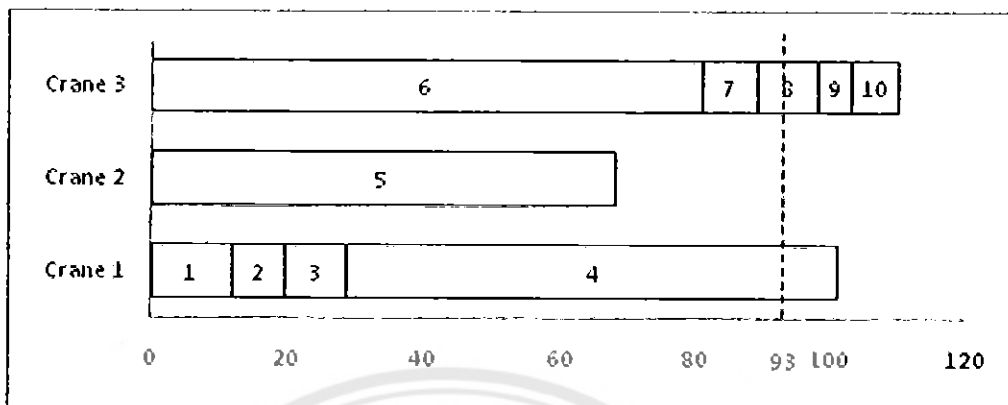
- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนเครื่อง)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อเครื่อง ( $Avg = \sum X / K$ ) เมื่อ X คือ Processing Time และ K คือ จำนวนเครื่อง
- 4) นับจำนวน Processing Time สะสมจากซ้ายไปขวา จะพบว่าค่าเฉลี่ยอาจจะเท่ากับค่าสะสมค่าใดค่าหนึ่งให้ตัดกลุ่มที่จำนวนนั้น หรืออยู่ระหว่างค่าสะสม 2 ค่า ในกรณีที่อยู่ระหว่างค่า 2 ค่า ให้พิจารณาว่าค่าใดอยู่ห่างจากค่าเฉลี่ยน้อยที่สุด ให้ตัดแบ่งกลุ่มที่งานนั้น
- 5) เริ่มนับจำนวน processing Time สะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนได้จำนวนกลุ่มเท่ากับจำนวนเครื่อง - 1 ให้เครื่องตัวสุดท้ายทำงานที่เหลือ

แผนผังการทำงานของโปรแกรม Greedy 5



รูปที่ 4.9 แผนผังการทำงานของโปรแกรม Greedy 5

แผนผังตัวอย่างการทำงานในแต่ละเครนของโปรแกรม Greedy 5



รูปที่ 4.10 แสดงการทำงานในแต่ละเครนของ Greedy 5 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.10 อธิบายการคำนวณตามวิธี Greedy 5 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่าง โจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68,81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 5 คือการคำนวณให้แต่ละเครนมีการทำงานโดยใกล้เคียงกับเฉลี่ยมากที่สุดมีวิธีการทำงานคือ เครนที่ 1 เลือกทำงานที่ 1 จากนั้นทำการบวกรงานที่ 2 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรงานที่ 3 เพิ่มยังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรงานที่ 4 เข้าไป ตรวจผลลัพธ์ปรากฏว่าได้ค่าคือ 101 ทำการบวกรงานที่ 5 เข้ามาได้ค่า 169 พบว่า 101 ใกล้เคียงกับ 93 มากที่สุดดังนั้น เครนที่ 1 หยุดทำงานที่งานที่ 4 เริ่มการคำนวณค่าของเครนที่ 2 โดยเริ่มที่งานที่ 5 มีค่า 68 จากนั้นทำการบวกรงานที่ 6 เพิ่มปรากฏว่าได้ค่า 149 พบว่า 68 ใกล้เคียง 93 มากกว่า หยุดการทำงานของเครนที่ 2 ที่งานที่ 5 จึงทำการเริ่มคำนวณค่าของเครนที่ 3 โดยนำงานที่เหลือมารวมกัน โดยมีค่าเวลาการทำงานสูงที่สุดที่เครน 3 ใช้เวลา 110 วินาที

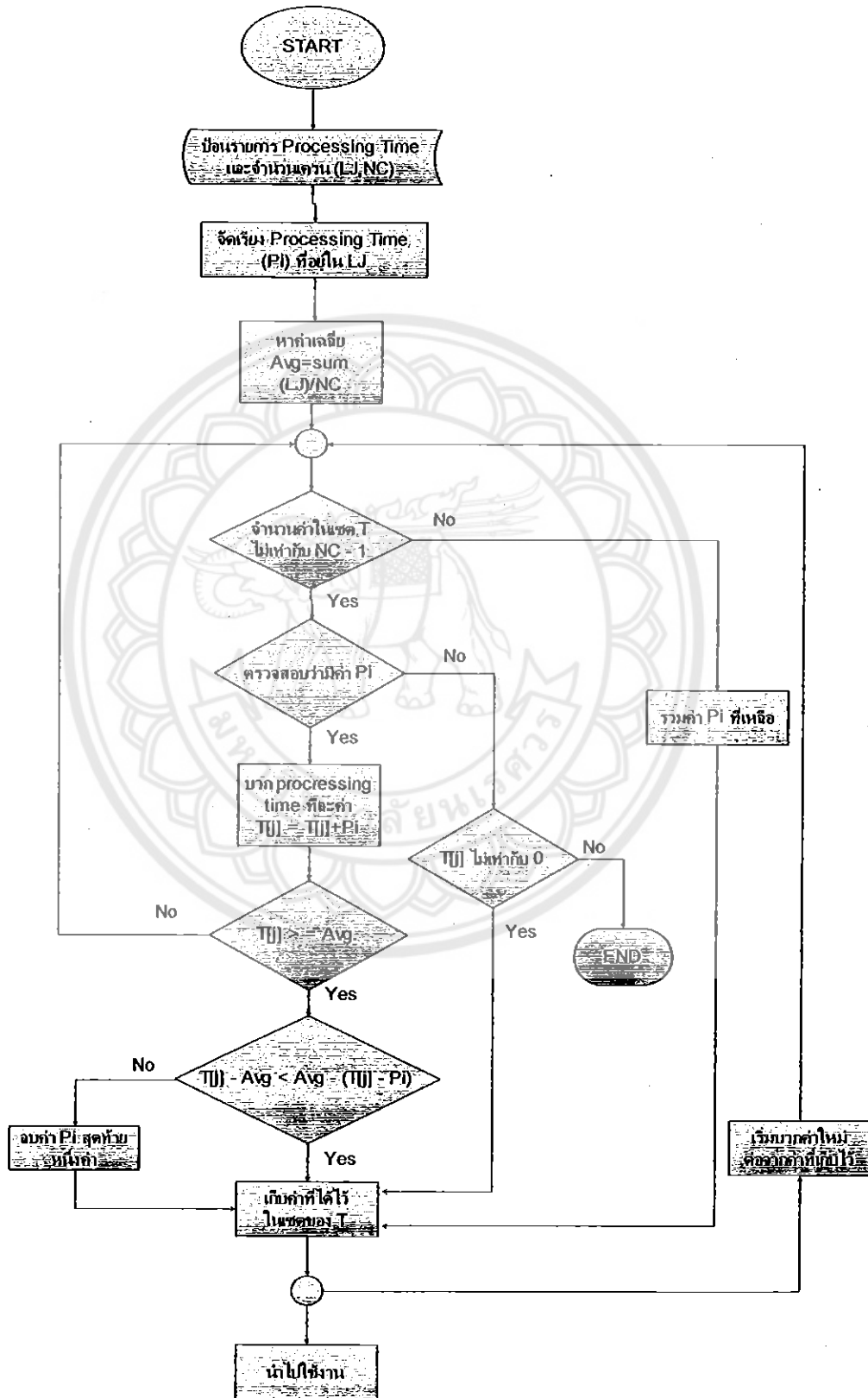
#### 4.1.6 วิธี Greedy 6

โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่าภาระการทำงานของครนนั้นสามารถหาค่าได้โดย การปล่อยให้ครนที่ 1 ทำงานตามลำดับด้วยปัญหาที่สร้างขึ้น และเก็บค่าของปัญหานั้น ย้อนหลัง โดยเมื่อเกินค่าเฉลี่ยของปัญหาทั้งหมดและนำมาเปรียบเทียบกับค่าของปัญหาก่อนถึงค่าเฉลี่ยแล้วจึงทำการเลือกการทำงานที่ใกล้ค่าเฉลี่ยมากที่สุด ก่อนจะเริ่มทำงานที่ครนลำดับถัดไป โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

ขั้นตอนการทำงานของ โปรแกรม Greedy 6

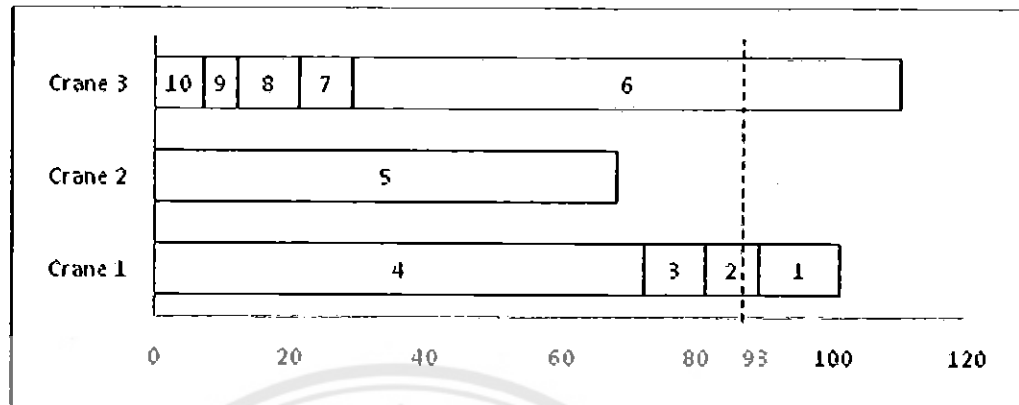
- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม (เท่ากับจำนวนครน)
- 3) หาค่าเฉลี่ยของ Processing Time ต่อครน ( $Avg = \sum X / K$ ) เมื่อ X คือ Processing Time และ K คือ จำนวนครน
- 4) นับจำนวน Processing Time สะสมจากขวาไปซ้าย จะพบว่าค่าเฉลี่ยอาจจะเท่ากับค่าสะสมค่าใดค่าหนึ่งให้ตัดกลุ่มที่จำนวนนั้น หรืออยู่ระหว่างค่าสะสม 2 ค่า ในกรณีที่อยู่ระหว่างค่า 2 ค่า ให้พิจารณาว่าค่าใดอยู่ห่างจากค่าเฉลี่ยน้อยที่สุด ให้ตัดแบ่งกลุ่มที่งานนั้น
- 5) เริ่มนับจำนวน processing Time สะสมใหม่ โดยเริ่มนับต่อจากกลุ่มก่อนหน้า
- 6) กลับไปทำตามข้อ 4 และ 5 ไปเรื่อย ๆ จนได้จำนวนกลุ่มเท่ากับจำนวนครน - 1 ให้ครนตัวสุดท้ายทำงานที่เหลือ

แผนผังการทำงานของ โปรแกรม Greedy 6



รูปที่ 4.11 แผนผังการทำงานของ โปรแกรม Greedy 6

### แผนผังตัวอย่างการทำงานในแต่ละเครื่องของโปรแกรม Greedy 6



รูปที่ 4.12 แสดงการทำงานในแต่ละเครื่องของ Greedy 6 โดยใช้ตัวอย่างปัญหางานที่ 6 ขนาดกลาง

จากรูปที่ 4.12 อธิบายการคำนวณตามวิธี Greedy 6 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68, 81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 6 คือการคำนวณให้แต่ละเครื่องมีการทำงานที่ใกล้เคียงค่าเฉลี่ยที่สุด โดยเริ่มทำงานจากลำดับสุดท้ายมาลำดับแรกมีวิธีการทำงานคือ เครื่องที่ 3 เลือกทำงานที่ 10 จากนั้นทำการบวกรงานที่ 9 เข้ามา ทำการตรวจว่าผลรวมมีค่าเกินกว่าค่าเฉลี่ยหรือไม่ ในกรณีนี้ยังไม่เกินค่าเฉลี่ย จึงทำการบวกรงานที่ 8 เพิ่มยังไม่ถึงค่าเฉลี่ยที่ 93 ทำการบวกรงานที่ 7 เข้าไปตรวจผลลัพธ์ปรากฏว่ายังไม่มากกว่าค่าเฉลี่ย บวกรงานที่ 6 เข้าไปได้ผลลัพธ์คือ 110 จากนั้นนำงานที่ 5 บวกเข้าได้ค่าคือ 178 พบว่า 110 ใกล้ถึง 93 มากกว่า หยุดการทำงานของเครื่องที่ 3 ที่งานที่ 6 เริ่มการคำนวณค่าของเครื่องที่ 2 โดยเริ่มที่งานที่ 5 ได้ผลลัพธ์ 68 จากนั้นทำการบวกรงานที่ 4 เพิ่ม ปรากฏว่าได้ผลลัพธ์คือ 140 พบว่า 68 ใกล้ถึง 93มากกว่า หยุดการทำงานของเครื่องที่ 2 ที่งานที่ 4 จึงทำการเริ่มคำนวณค่าของเครื่องที่ 1 โดยนำงานที่เหลือมารวมกัน พบว่ามีค่าเวลาการทำงานสูงที่สุดที่เครื่อง 3 ใช้เวลา 110 วินาที

#### 4.1.7 วิธี Greedy 7

โปรแกรมนี้ถูกสร้างขึ้นด้วยแนวคิดที่ว่า การจะหาการทำงานของแต่ละเครื่องนั้นสามารถหาค่าได้โดย การให้เครื่องทำงาน โดยการสุ่มค่าที่ดีที่สุดขึ้นมา โดยในที่นี้จะใช้ค่าช่วงในการสุ่ม  $NS = 100000$  ค่า เพื่อหาค่าคำตอบที่ดีที่สุดออกมา โดยมีโครงสร้างโปรแกรมที่สร้างขึ้นดังนี้

### ขั้นตอนการทำงานของโปรแกรม Greedy 7

- 1) จัดเรียงลำดับงานของ Processing Time ตามลำดับจากซ้ายไปขวา
- 2) กำหนดจำนวนกลุ่ม Processing Time (เท่ากับจำนวนเครน)
- 3) กำหนดจำนวนครั้งในการ random
- 4) ทำการ random โดยการตัดข้อมูลเป็นช่วง ๆ ให้มีจำนวนกลุ่มเท่ากับจำนวนเครน
- 5) รวมข้อมูลของแต่ละช่วงของทุกค่าที่ Random
- 6) เลือกค่า max จากช่วงของแต่ละกลุ่มที่ Random
- 7) เลือกค่า min จากขั้นตอน 6
- 8) ตรวจสอบว่าอยู่ในการ random ครั้งใด และมีผลการแบ่งแบบใด



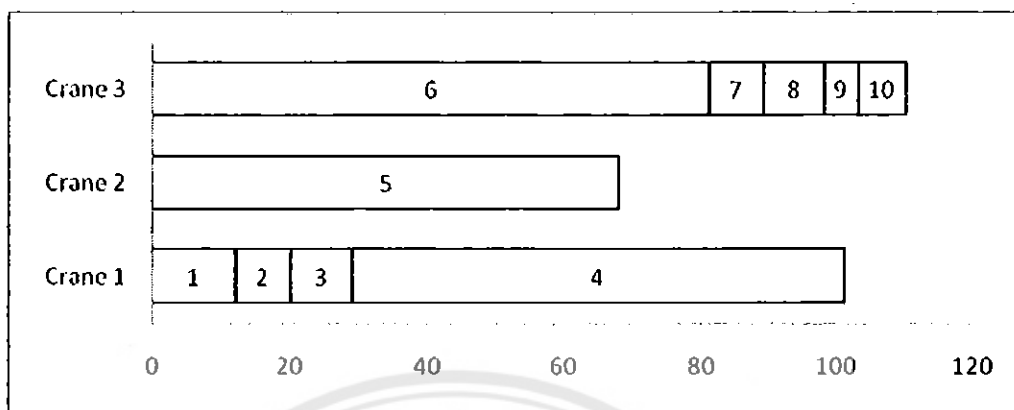


## แผนผังการทำงานของโปรแกรม Greedy 7



รูปที่ 4.13 แผนผังการทำงานของโปรแกรม Greedy 7

แผนผังตัวอย่างการทำงานในแต่ละเครื่องของโปรแกรม Greedy 7



รูปที่ 4.14 แสดงการทำงานในแต่ละเครื่องของ Greedy 7 โดยใช้ตัวอย่างปัญหาขนาดกลาง

จากรูปที่ 4.14 อธิบายการคำนวณตามวิธี Greedy 7 แบบฮิวริสติกส์ได้ดังนี้

ใช้ตัวอย่างโจทย์ปัญหาแบบที่ 6 ปัญหาขนาดกลาง มีข้อมูลของ Processing Time คือ 12,8,9,72,68, 81,8,9,5,7 โดยนับตามลำดับ เช่นงานที่ 1 ใช้เวลาในการทำงาน 12 วินาที งานที่ 2 ใช้เวลาในการทำงาน 8 วินาที เป็นต้น ในที่นี้ Greedy 7 คือการใช้วิธีการสุ่มหาคำตอบที่ดีโดยกำหนดให้มีค่าการสุ่มอยู่ที่ 100,000 ครั้ง วิธีนี้เป็นการแบ่งกลุ่มของปัญหาแบบสุ่ม ให้มีจำนวนกลุ่มในแต่ละปัญหาเท่ากับจำนวนของเครื่อง โดยค่าในแต่ละกลุ่มจะต้องมีลำดับติดกัน แล้วทำการสุ่มตามจำนวนครั้งที่ได้กำหนดไว้ เมื่อเราทำการสุ่มครบแล้ว ทำการรวมค่าในแต่ละกลุ่มจากนั้นเลือกค่าที่มากที่สุดของแต่ละครั้งที่สุ่มออกมา เมื่อได้แล้วให้เลือกค่าน้อยที่สุด แล้วตรวจสอบว่ามาจากกลุ่มไหน นั่นก็คือคำตอบของวิธีนี้ ซึ่งจากตัวอย่างปัญหาที่คำนวณ โปรแกรมได้คำนวณการทำงานได้คำตอบว่า มีการแบ่งกลุ่มแบบ 4,5,10 หมายความว่า งาน 1-4 ทำงานโดยเครื่องที่ 1 ใช้เวลาทำงาน 101 วินาที งาน 5 ถูกทำโดยเครื่องที่ 2 ใช้เวลาทำงาน 68 วินาที และงานที่ 6-10 ถูกทำงานโดยเครื่องที่ 3 ใช้เวลาทำงาน 110 วินาที

## 4.2 ปัญหางานที่ใช้ในการคำนวณ

สำหรับปัญหาของงานนี้ กลุ่มผู้วิจัยจะใช้ทั้งปัญหางานแบบที่กลุ่มวิจัยของเอราวิลและคณะใช้ในการประมวลผลบนโปรแกรมสำเร็จรูป และปัญหางานใหม่ที่มีความหลากหลายเพื่อเพื่อนำมาทดสอบ

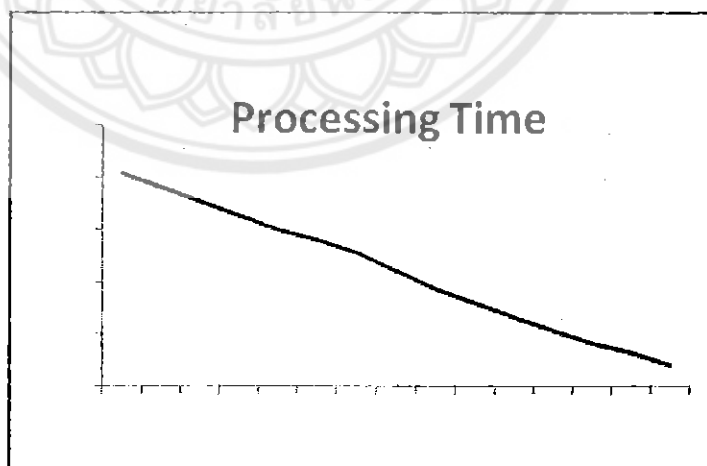
### 4.2.1 ขนาดของปัญหางาน

เนื่องจากงานที่ถูกทำครบในแต่ละครั้งมีจำนวนงานไม่เท่ากัน อาจขึ้นอยู่กับ คำสั่งนำเข้า ขนาดเรือ เป็นต้น จึงต้องมีการแจกแจงจำนวนของงาน ซึ่งเราแบ่งปัญหาออกเป็น 3 ขนาด คือ

- 1) ปัญหาขนาดเล็ก (มีจำนวนงานประมาณ 1 - 6 งาน )
- 2) ปัญหาขนาดกลาง (มีจำนวนงานประมาณ 6 - 12 งาน )
- 3) ปัญหาขนาดใหญ่ (มีจำนวนงานประมาณ 12 งานขึ้นไป )

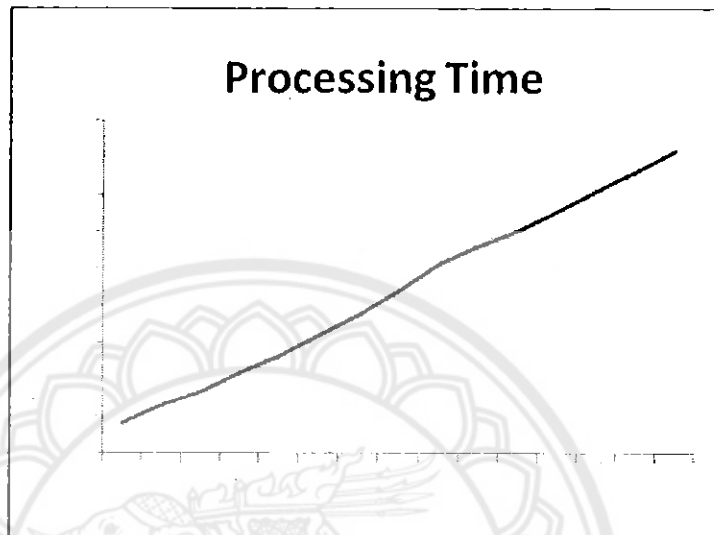
### 4.2.2 ลักษณะของงาน

เนื่องจากงานแต่ละงานใช้เวลาในการขนย้ายไม่เท่ากัน จึงต้องมีการจำลองเวลางานในลักษณะต่างๆเพื่อใช้ในการประมวลผลบนโปรแกรม Python และบน โปรแกรมสำเร็จรูปแบบที่ 1 ข้อมูลเรียงจาก Processing Time มากไปน้อย โดยให้ชุดข้อมูลลำดับแรกมีคเวลาการทำงานที่มากและมีแนวโน้มลดลงดังกราฟต่อไปนี้



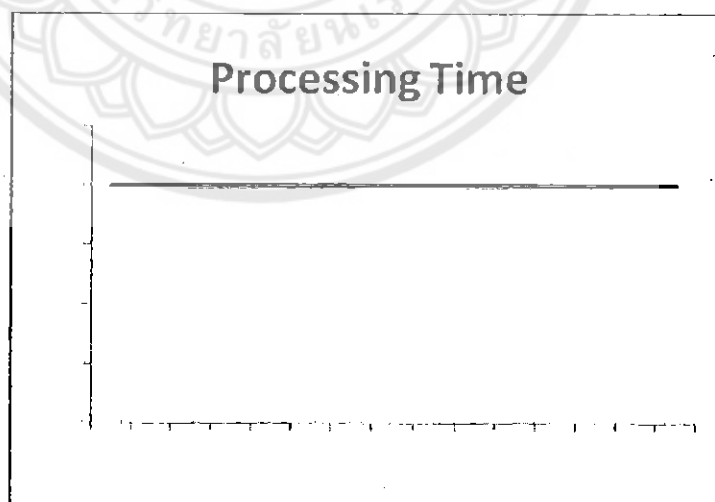
รูปที่ 4.15 กราฟแสดงลักษณะของงานแบบที่ 1

แบบที่ 2 ข้อมูลเรียงจาก Processing Time น้อยไปมาก โดยให้ชุดข้อมูลลำดับแรกมีค่าเวลาการทำงานที่น้อยและมีแนวโน้มเพิ่มขึ้นดังกราฟต่อไปนี้



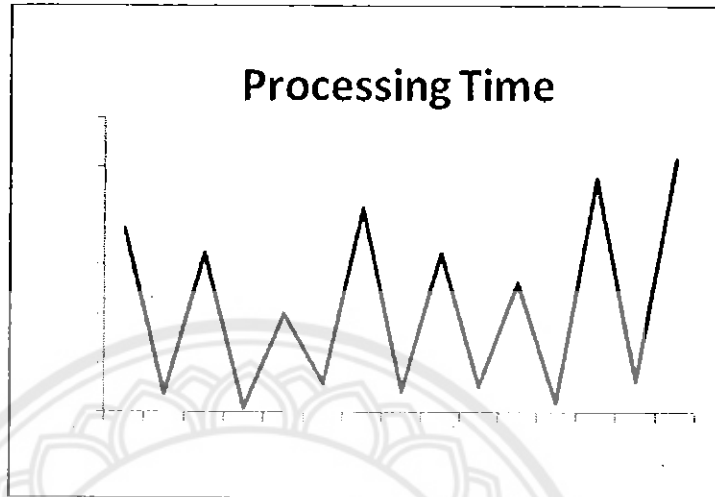
รูปที่ 4.16 กราฟแสดงลักษณะของงานแบบที่ 2

แบบที่ 3 ปัญหาที่มี Processing Time ใกล้เคียงกัน โดยให้ชุดข้อมูลมีค่าเวลาการทำงานที่คงที่ดังกราฟต่อไปนี้



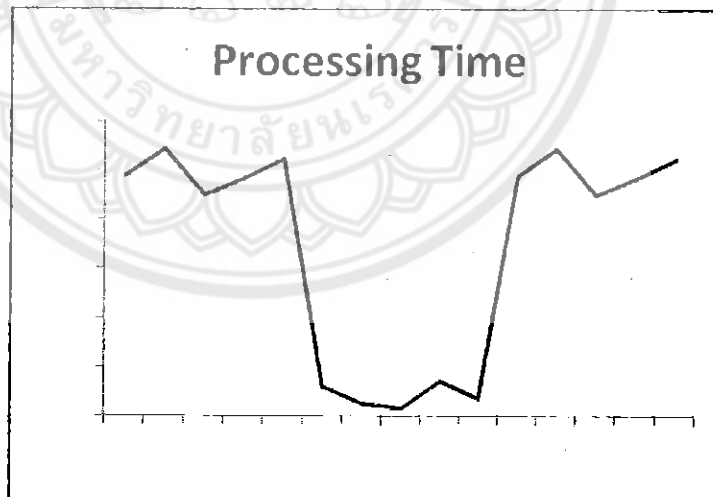
รูปที่ 4.17 กราฟแสดงลักษณะของงานแบบที่ 3

แบบที่ 4 ปัญหาความต่างของ Processing Time มาก โดยให้ชุดข้อมูลมีค่าเวลาที่ใกล้เคียงกันมีค่าต่างกันมาก โดยมีแนวโน้มของชุดข้อมูลดังกราฟต่อไปนี้



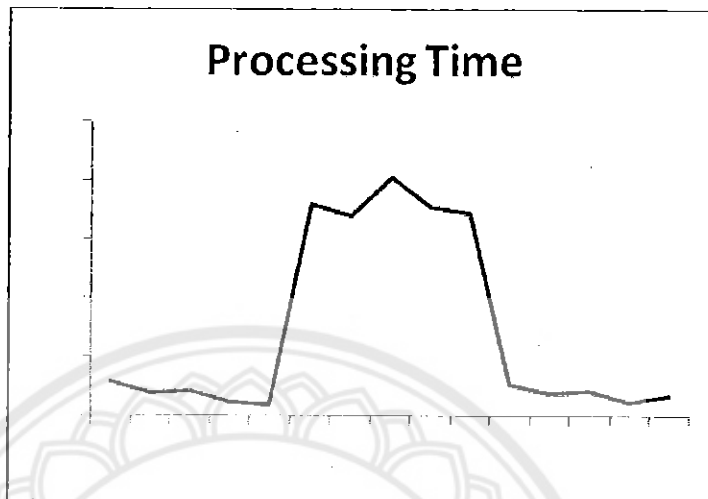
รูปที่ 4.18 กราฟแสดงลักษณะของงานแบบที่ 4

แบบที่ 5 ปัญหา Processing Time มากช่วงต้นและท้ายของปัญหา โดยให้ชุดข้อมูลมีค่าเวลาการทำงานที่น้อยที่สุดในช่วงกลางของชุดข้อมูล โดยมีแนวโน้มดังกราฟต่อไปนี้



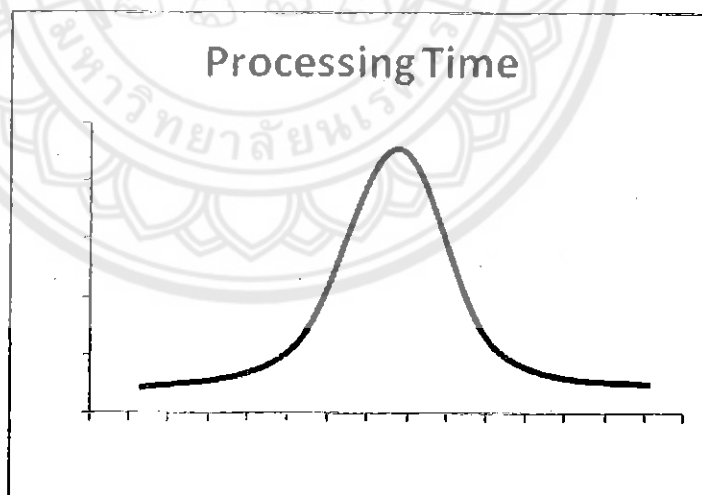
รูปที่ 4.19 กราฟแสดงลักษณะของงานแบบที่ 5

แบบที่ 6 ปัญหามี Processing Time มากช่วงกลางของปัญหา โดยให้ชุดข้อมูลมีค่าเวลาการทำงานที่มากที่สุดในช่วงกลางของชุดข้อมูล โดยมีแนวโน้มดังกราฟต่อไปนี้



รูปที่ 4.20 กราฟแสดงลักษณะของงานแบบที่ 6

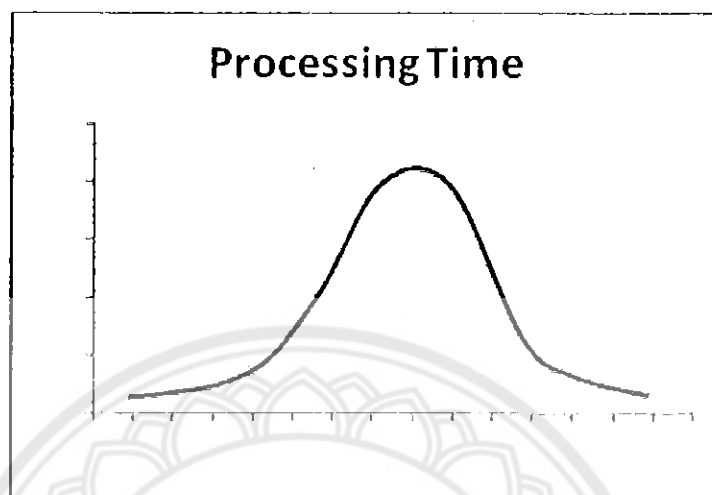
แบบที่ 7 Random ปัญหาด้วยวิธีการแจกแจงแบบ Normal (ใน Excel ใช้คำสั่ง Norminv) โดยให้ค่าเฉลี่ย=25 และมีค่าเบี่ยงเบน=2



รูปที่ 4.21 กราฟแสดงลักษณะของงานแบบที่ 7

แบบที่ 8 Random ปัญหาด้วยวิธีการแจกแจงแบบ Normal (ใน Excel ใช้คำสั่ง Norminv)

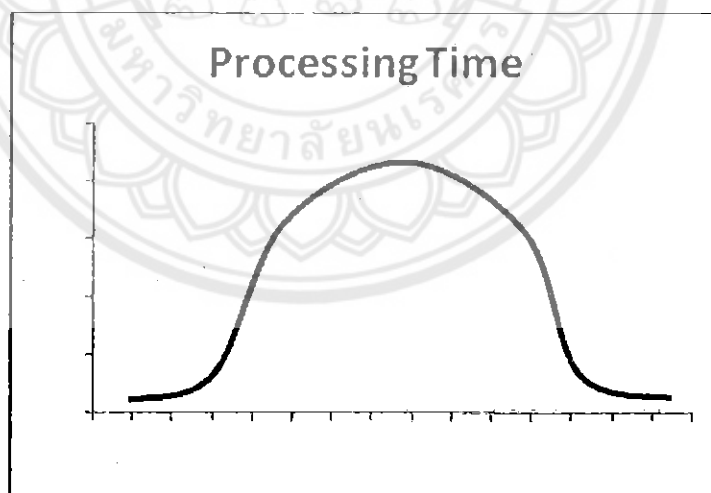
โดยให้ค่าเฉลี่ย=25 และมีค่าเบี่ยงเบน=10



รูปที่ 4.22 กราฟแสดงลักษณะของงานแบบที่ 8

แบบที่ 9 Random ปัญหาด้วยวิธีการแจกแจงแบบ Normal (ใน Excel ใช้คำสั่ง Norminv)

โดยให้ค่าเฉลี่ย=25 และมีเบี่ยงเบน=25



รูปที่ 4.23 กราฟแสดงลักษณะของงานแบบที่ 9

จากการสุ่มปัญหาทางานเพื่อนำมาใช้ในการทดสอบผลการทดลองทางผู้วิจัยได้ทำการสุ่มปัญหา  
งานขึ้นตามรูปแบบที่ได้กล่าวไปแล้ว 9 แบบ โดยได้ชุดข้อมูลโดยเป็นเวลาการทำงานในแต่ละงาน  
กับลำดับของงานตามตารางข้างล่างต่อไปนี้

ตารางที่ 4.1 ปัญหางานในขนาดและลักษณะต่าง ๆ

แบบ ลักษณะ ของงาน	ขนาดของ ปัญหา	จำนวน เครน	Processing Time (sec.)
1	เล็ก	3	409,382,355,328,301
	กลาง	3	409,382,355,328,301,280,255,220,187,159
	ใหญ่	3	409,382,355,328,301,280,255,220,187,159,131,108,82,65,40
2	เล็ก	3	40,65,82,108,131
	กลาง	3	40,65,82,108,131,159,187,220,255,280
	ใหญ่	3	40,65,82,108,131,159,187,220,255,280,301,328,355,382,409
3	เล็ก	3	20,20,20,20,20
	กลาง	3	20,20,20,20,20,20,20,20,20,20
	ใหญ่	3	20,20,20,20,20,20,20,20,20,20,20,20,20,20,20
4	เล็ก	3	75,8,65,2,40
	กลาง	3	75,8,65,2,40,12,83,9,65,11
	ใหญ่	3	75,8,65,2,40,12,83,9,65,11,53,4,95,13,103
5	เล็ก	3	109,90,5,3,109
	กลาง	3	98,109,90,5,3,14,109,90,97,105
	ใหญ่	3	98,109,90,97,105,12,5,3,14,7,98,109,90,97,105
6	เล็ก	3	8,9,72,68,5
	กลาง	3	12,8,9,72,68,81,8,9,5,7
	ใหญ่	3	12,8,9,5,4,72,68,81,71,69,11,8,9,5,7
7	เล็ก	3	100,98,99,98,102
	กลาง	3	51,53,46,50,48,50,51,48,52,55
	ใหญ่	3	25,27,30,27,27,24,24,25,26,22,27,28,27,27,25
8	เล็ก	3	99,99,104,104,118
	กลาง	3	37,40,42,44,49,54,58,62,66,78
	ใหญ่	3	3,18,18,19,20,22,22,23,24,23,26,34,40,40,42
9	เล็ก	3	91,91,138,141,277
	กลาง	3	14,1,12,20,47,72,91,112,129,190
	ใหญ่	3	31,10,8,9,18,20,22,28,28,37,41,46,62,62,66



ตารางแสดงปัญหาหางานกลุ่มวิจัยของเอราวิลได้ถูกนำมาประมวลผลด้วยโปรแกรม PQCS ด้วยเพื่อนำผลที่ได้มาเปรียบเทียบกับการทำงานของโปรแกรมที่สร้างขึ้นโดยมีค่าดังตารางต่อไปนี้

**ตารางที่ 4.2 ปัญหาหางานของกลุ่มวิจัยของเอราวิลและคณะ**

No.	Crane	Job	Processing Time (sec.)
1	3	5	2,3,4,5,3
2	3	5	15,17,23,12,10
3	3	6	13,32,22,17,25,19
4	3	7	36,24,24,41,35,33,15
5	3	6	5,1,2,8,1,2
6	3	8	15,12,27,28,11,22,41,35
7	5	9	4,4,4,3,3,7,2,2,7
8	5	6	11,22,10,10,10,6
9	5	7	1,2,4,1,2,1,1
10	5	8	25,25,35,25,25,35,25,25
11	5	10	26,18,26,30,18,55,30,15,30,15
12	5	12	20,22,25,20,29,21,25,29,25,23,24,27

#### 4.3 ผลการทดสอบ

เพื่อทดสอบการประมวลผลด้านผลลัพธ์และเวลาบนโปรแกรม Python และโปรแกรมสำเร็จรูปของปัญหาหางานทั้ง 2 กลุ่มคือปัญหาหางานที่สร้างขึ้นใหม่และปัญหาหางานของเอราวิลและคณะ โดยได้ผลดังตารางต่อไปนี้



ตารางที่ 4.4 ตารางแสดงผลลัพธ์การประมวลผลปัญหาทางตามวิธีเฮอราวิลและคะแนนของแบบจำลองชีวิตศึกษาโปรแกรม Python

หมายเลข โจทย์		จำนวน ข้อ		Python												PC Run time (sec)											
				G1			G2			G3			G4				G5			G6			G7				
				Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)			Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)		Imax (sec)	Run Time (sec)
1	3-5	9	0	8	0	5	0	9	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8	2.69
2	3-5	35	0	45	0	45	0	32	0	32	0	32	0	32	0	32	0	32	0	32	0	32	0	32	0	32	2.56
3	3-6	64	0	71	0	83	0	67	0	45	0	45	0	45	0	45	0	45	0	45	0	45	0	45	0	45	2.56
4	3-7	84	0	89	0	83	0	135	0	83	0	83	0	83	0	83	0	83	0	83	0	83	0	83	0	83	2.58
5	3-6	8	0	11	0	11	0	5	0	10	0	10	0	10	0	10	0	10	0	10	0	10	0	10	0	10	2.61
6	3-8	52	0	88	0	76	0	93	0	76	0	76	0	76	0	76	0	76	0	76	0	76	0	76	0	76	2.59
7	5-9	11	0	11	0	18	0	17	0	17	0	17	0	17	0	17	0	17	0	17	0	17	0	17	0	17	4.05
8	5-6	33	0	22	0	22	0	35	0	22	0	22	0	22	0	22	0	22	0	22	0	22	0	22	0	22	3.9
9	5-7	4	0	5	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	4
10	5-8	60	0	60	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	110	3.92
11	5-10	103	0	85	0	90	0	70	0	70	0	70	0	70	0	70	0	70	0	70	0	70	0	70	0	70	3.94
12	5-10	79	0	79	0	89	0	87	0	87	0	87	0	87	0	87	0	87	0	87	0	87	0	87	0	87	3.95

จากการประมวลผลโดยใช้ปัญหางานแบบใหม่ที่สร้างขึ้นบนโปรแกรมสำเร็จรูป ได้ผลลัพธ์ดังตารางข้างล่างนี้

**ตารางที่ 4.5** ตารางแสดงผลลัพธ์การประมวลผลปัญหางานที่สุ่มขึ้นใหม่ของแบบจำลองฮิวริสติกส์บนโปรแกรมสำเร็จรูป

งาน		สำเร็จรูป	
		เอราวิล	
ชนิดของงาน	ขนาดของงาน	Tmax (sec.)	Run Time (sec.)
1	เล็ก	710	0
	กลาง	1017	>36000
	ใหญ่	-	>36000
2	เล็ก	173	1
	กลาง	522	>36000
	ใหญ่	-	>36000
3	เล็ก	40	0
	กลาง	80	>36000
	ใหญ่	100	>36000
4	เล็ก	75	0
	กลาง	140	569
	ใหญ่	-	>36000
5	เล็ก	109	0
	กลาง	285	>36000
	ใหญ่	-	>36000
6	เล็ก	73	0
	กลาง	97	1380
	ใหญ่	-	>36000
7	เล็ก	198	1
	กลาง	192	>36000
	ใหญ่	134	>36000
8	เล็ก	203	1
	กลาง	184	>36000
	ใหญ่	-	>36000
9	เล็ก	277	1
	กลาง	253	>36000
	ใหญ่	-	>36000

จากการประมวลผล โดยใช้ปัญหาทางานตามวิจัยของเอราวิตและคณะบนโปรแกรมสำเร็จรูป ได้ผลลัพธ์ดังตารางข้างล่างนี้

**ตารางที่ 4.6** ตารางแสดงผลลัพธ์การประมวลผลปัญหาทางานตามวิจัยเอราวิตและคณะของแบบจำลองฮิวริสติกส์บนโปรแกรมสำเร็จรูป

งาน		สำเร็จรูป	
		เอราวิต	
ชนิดของงาน	ขนาดของงาน-เครน	Tmax (sec.)	Run Time (sec.)
1	3--5	8	1
2	3--5	32	1
3	3--6	45	2
4	3--7	77	43
5	3--6	8	1
6	3--8	68	37
7	5--9	9	19
8	5--6	22	1
9	5--7	4	1
10	5--8	50	4
11	5--10	66	1440
12	5--12	68	>36000

#### 4.4 อธิบายผลการรันโปรแกรม Python ที่ได้จากการทดลองโปรแกรม G1-G6 และ G7

ตัวอย่างผลการทดลองของโปรแกรม G1-G6

```
C:/Python26/pythonw.exe -u "C:/Documents and Settings/Windows XP/My Documents/G1-7/G1.py"

Greedy 1

Average Time for each crane is 162.666666667

List of Total Time for each group is [174, 186, 128]

List of crane assigned to each job is [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3]

Running Time is 0.0 Seconds
```

"C:/Python26/pythonw.exe -u "C:/Documents and Settings/Windows XP/My Documents/G1-7/G1.py"" คือ ที่อยู่ของ โปรแกรมที่เซฟไว้

"Greedy 1" คือ ชื่อของ โปรแกรม

"Average Time for each crane is 162.666666667" คือ ค่าเฉลี่ยของ Processing Time ทั้งหมดต่อจำนวนเครน

"List of Total Time for each group is [174, 186, 128]" คือ บอกว่าเครนแต่ละตัวใช้เวลาทำงานรวมเท่าไร

"List of crane assigned to each job is [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3]" คือ บอกว่าเครนไหนใช้ทำงานใด เช่นในตัวอย่างบ่งบอกว่า งาน 1-9 ถูกทำงานโดยเครนที่ 1, งาน 10-13 ถูกทำงานโดยเครนตัวที่ 2 งาน 14และ15 ถูกทำงาน โดยเครนตัวที่ 3

"Running Time is 0.0 Seconds" คือ เวลาที่ใช้ในการประมวลผล

## ตัวอย่างผลการทดลองของ G7

```

C:/Python26/pythonw.exe -u "C:/Documents and Settings/Windows XP/My Documents/G1-
7/G7.py"

Greedy 7

The best solution is: 186

found at: 60

The solution is [174, 186, 128]

Each crane process as follows [9, 13, 15]

Running Time is 2.70300006866 Seconds

```

“C:/Python26/pythonw.exe -u "C:/Documents and Settings/Windows XP/My Documents/G1-7/G7.py"” คือ ที่อยู่ของโปรแกรมที่เซฟไว้

“Greedy 7” คือ ชื่อของ โปรแกรม

“The best solution is: 186” คือ เวลาที่ใช้ น้อยที่สุดในการทำงาน

“found at: 60” คือ ครั้งที่เจอค่าที่ดีที่สุดในการสุ่ม

“The solution is [174, 186, 128]” คือ บอกว่าเครนแต่ละตัวใช้เวลาทำงานรวมเท่าไร

“Each crane process as follows [9, 13, 15]” คือ บอกว่าเครนไหนใช้ทำงานใด เช่นในตัวอย่างบ่งบอกว่า งาน 1-9 ถูกทำงาน โดยเครนตัวที่ 1, งาน 10-13 ถูกทำงาน โดยเครนที่ 2 งาน 14และ15 ถูกทำงาน โดยเครนตัวที่ 3

“Running Time is 2.70300006866 Seconds” คือ เวลาที่ใช้ในการประมวลผล

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

หลังจากที่เราได้ผลการทดลองจากปัญหาตัวอย่างจากบทที่ 4 นั้น ในบทนี้จะทำการวิเคราะห์ และสรุปผลการทดลองที่ได้เพื่อเปรียบเทียบความแตกต่างระหว่างการประมวลผลบน โปรแกรม สำเร็จรูป และ โปรแกรม PQCS ซึ่งจะมีเนื้อหาดังต่อไปนี้

#### 5.1 วิเคราะห์ผลการทดลอง

เพื่อวัดประสิทธิภาพของแบบจำลองใหม่ที่ได้อัจฉริยะว่ามีประสิทธิภาพมากเพียงใด จึงได้นำมา เปรียบ เทียบกับผลการทดสอบแบบจำลองของเอราวิล หนึ่งฤทัย กนกพร ที่ประมวลผลโดย โปรแกรม สำเร็จรูป โดยใช้ปัญหางานที่มีจำนวนของครน งาน และเวลาของงานแต่ละงานเดียวกัน ซึ่งปัญหาของงานนี้จะมีทั้งปัญหาเก่าและปัญหาใหม่ที่ได้อัจฉริยะ ในการเปรียบเทียบนี้จะ เปรียบเทียบในด้านของเวลาเสร็จสิ้นในการทำงาน เวลาในการประมวลผล สถานะของคำตอบ ดังนี้



**ตารางที่ 5.1** ผลเปรียบเทียบการทำงานของโปรแกรมประมวลผลบนโปรแกรมสำเร็จรูป กับ โปรแกรม Python โดยใช้ปัญหาของเอราวิลและคณะ

Problem	โปรแกรมสำเร็จรูป			Python			%GAP
	Run time (sec.)	Results (sec.)	State	Name	Run time (sec.)	Results (sec.)	
P10.1	1	8	Optimum	G2	0	8	0.00 %
P10.2	1	32	Optimum	G4	0	32	0.00 %
P10.3	2	45	Optimum	G5	0	45	0.00 %
P10.4	43	77	Optimum	G5	0	83	7.79 %
P10.5	1	8	Optimum	G1	0	8	0.00 %
P10.6	37	68	Optimum	G3	0	76	11.76 %
P10.7	19	9	Optimum	G7	4.05	9	0.00 %
P10.8	1	22	Optimum	G2	0	22	0.00 %
P10.9	1	4	Optimum	G1	0	4	0.00 %
P10.10	4	50	Optimum	G5	0	50	0.00 %
P10.11	1440	66	Optimum	G4	0	70	6.06 %
P10.12	>36000	68	Feasible	G7	3.95	70	2.94 %

### 5.1.1 วิเคราะห์ผลการทดลองโดยใช้ปัญหาดามงานวิจัยของเอราวิลและคณะ

จากตารางผลการทดสอบจะเห็นว่า โปรแกรมสำเร็จรูป เป็นโปรแกรมที่สามารถหาคำคำตอบได้ดีกว่า แต่เวลาที่ใช้ในการหาผลลัพธ์นั้นจะขึ้นอยู่กับขนาดของปัญหา ยิ่งปัญหามีขนาดที่ใหญ่ขึ้น จำนวนการแตกสาขายิ่งมากขึ้นไปด้วยทำให้ไม่สามารถประมาณเวลาในการหาคำตอบที่ดีที่สุดได้ แต่โปรแกรม Python ที่เขียนขึ้นด้วยวิธีฮิวริสติกส์นั้น สามารถหาคำผลลัพธ์ได้ในเวลาที่สั้นแต่คำตอบที่ได้นั้นจะมีผลลัพธ์สูงกว่าโปรแกรมสำเร็จรูป อยู่บ้างตาม โจทย์ปัญหานี้ ซึ่งมีค่าความแตกต่างกันเฉลี่ย 2.34% และมากที่สุดเท่ากับ 11.76% ซึ่งถือว่ามีค่าใกล้เคียงกันแต่ทดแทนจุดค่อยด้วยเวลาที่ใช้ในการประมวลผลที่รวดเร็วกว่า

ตารางที่ 5.2 ผลเปรียบเทียบการทำงานของโปรแกรมประมวลผลบนโปรแกรมสำเร็จรูป กับโปรแกรม Python โดยใช้ปัญหาที่สร้างขึ้น

Problem	โปรแกรมสำเร็จรูป			Python			%GAP
	Run time(sec.)	Results(sec.)	State	Name	Run time (sec.)	Results(sec.)	
1.1	0	710	Optimum	G2	0	737	3.80 %
1.2	>36000	1017	Feasible	G2	0	1101	8.26 %
1.3	>36000	-	Unknow	G1	0	1164	- %
2.1	1	173	Optimum	G4	0	187	8.09 %
2.2	>36000	522	Feasible	G2	0	566	8.43 %
2.3	>36000	-	Unknow	G2	0	1164	- %
3.1	0	40	Optimum	G1	0	40	0.00 %
3.2	>36000	80	Feasible	G1	0	80	0.00 %
3.3	>36000	100	Feasible	G1	0	100	0.00 %
4.1	0	75	Optimum	G1	0	75	0.00 %
4.2	569	140	Optimum	G1	0	148	5.71 %
4.3	>36000	-	Feasible	G2	0	221	- %
5.1	0	109	Optimum	G5	0	109	0.00 %
5.2	>36000	285	Feasible	G3	0	292	2.46 %
5.3	>36000	-	Unknow	G1	0	394	- %
6.1	0	73	Optimum	G2	0	73	0.00 %
6.2	1380	97	Optimum	G5	0	110	13.40 %
6.3	>36000	-	Unknow	G1	0	178	- %
7.1	1	198	Optimum	G1	0	198	0.00 %
7.2	>36000	192	Feasible	G7	2.56	199	3.65 %
7.3	>36000	134	Feasible	G5	0	136	1.49 %
8.1	1	203	Optimum	G1	0	208	2.46 %
8.2	>36000	184	Feasible	G2	0	206	11.96 %
8.3	>36000	-	Unknow	G5	0	130	- %
9.1	1	277	Optimum	G2	0	279	0.72 %
9.2	>36000	253	Feasible	G1	0	257	1.58 %
9.3	>36000	-	Unknow	G1	0	186	- %

### 5.1.2 วิเคราะห์ผลการทดลองโดยใช้ปัญหาที่สร้างขึ้นใหม่

จากตารางแสดงผลการทดสอบข้างต้นจะเห็นว่าโปรแกรมสำเร็จรูป นั้นจะพัฒนา คำตอบที่ดีตามเวลาที่มากขึ้น ดังนั้นถ้าต้องการคำตอบในเวลาที่ย่ำกักจะไม่สามารถที่จะประมาณ เวลาล่วงหน้าได้ เพราะเราจะไม่สามารถทราบได้เลยว่าปัญหานั้นๆ จะต้องแตกสาขาเท่าไรและจะ เจอคำตอบที่ดีที่สุดเมื่อไร จากตาราง โปรแกรมสำเร็จรูป จะสังเกตเห็นว่าบางปัญหานั้นตัวโปรแกรม สำเร็จรูปเองไม่สามารถที่จะหาคำตอบได้เนื่องจากยังไม่สามารถที่จะคำนวณหาสาขาที่จะต้องแตก ได้เลยในเวลา 36,000 วินาที จึงไม่สามารถแม้ที่จะหาคำตอบออกมาได้ ส่วน โปรแกรม Python ที่ เขียน โดยใช้วิธีฮิวริสติกส์นั้น สามารถประมาณคำตอบที่ใกล้เคียงจนถึงดีที่สุดได้ในเวลาที่รวดเร็ว กว่า เหมาะกับการต้องการผลลัพธ์ที่มีเวลาจำกัด ในตาราง โปรแกรม Python จะเห็นว่าสามารถ ประมาณคำตอบได้ดีที่สุดในกรณีที่เป็นปัญหามานกลาง และ เวลาของปัญหาเป็นแบบเท่ากันอีกทั้ง ยังสามารถประมาณคำตอบของปัญหาที่มีขนาดใหญ่ได้ในเวลาอันสั้น

### 5.2 สรุปผลการทดลอง

จากการทดลองจะเห็นได้ว่า โปรแกรมสำเร็จรูป เป็นโปรแกรมที่สามารถหาคำคำตอบได้ดีที่สุด แต่ เนื่องจาก โปรแกรมที่ถูกสร้างขึ้นและประมวลผลบนโปรแกรมสำเร็จรูป นั้นใช้วิธีการหา คำตอบแบบ Branch and Bound ซึ่งจะต้องทำการแตกสาขาดังนั้นถ้าปัญหามานกลางหลายๆ เราจะไม่สามารถประมาณเวลาในการหาคำตอบที่ดีที่สุดได้เนื่องจากใช้เวลาในการประมวลผลที่ นาน ส่วนโปรแกรม PQCS ที่ผู้วิจัยสร้างขึ้นโดยใช้หลักการแบบฮิวริสติกส์นั้นสามารถประมาณ คำตอบที่เหมาะสมได้ในเวลาที่รวดเร็ว แต่คำตอบที่ได้นั้นมีค่ามากกว่าการประมวลผลบน โปรแกรมสำเร็จรูป อยู่เฉลี่ยที่ 3.14% ตามปัญหาทางานที่ได้ทำการทดลอง

เพราะฉะนั้นสรุปได้ว่า โปรแกรม Python ที่ถูกเขียนวิธีฮิวริสติกส์นั้นสามารถประมวลผลหา คำตอบได้เร็วกว่าการประมวลผลบนโปรแกรมสำเร็จรูป โดยเฉพาะปัญหาที่มีขนาดใหญ่นั้น PQCS สามารถประมาณคำตอบได้ในเวลาอันรวดเร็วแต่ผลที่ได้อาจจะมีย่ำผลลัพธ์มากกว่าการประมวลผล บนโปรแกรมสำเร็จรูป อยู่เล็กน้อย

## เอกสารอ้างอิง

คณะวิศวกรรมศาสตร์ (บางเขน). (2551). การวิจัยดำเนินงานสำหรับวิศวกร I. Branch and Bound.

สืบค้นเมื่อ 17 สิงหาคม 2551, จาก

<http://cyberlab.lh1.ku.ac.th/elearn/faculty/engineer/bkk42/Doc/Contents/08Branch&Bound.pdf>

เอราวิดี ถาวร, หนึ่งฤทัย ทัพไพใหญ่ และกนกพร อารยิกานนท์. (2550). การแก้ปัญหาการจัดลำดับงานของเครื่องโดยใช้โปรแกรมเชิงคณิตศาสตร์. พิษณุโลก : มหาวิทยาลัยนเรศวร

Vernon L.Ceder. (2010) . **The quick Python book**. (2<sup>nd</sup>ED). New York: Manning Publications Co.





## โปรแกรม PQCS

Python

```
def G1(LJ,NC):  
    print 'Greedy 1'  
    Total = sum(LJ)  
    Avg = float(Total)/NC  
    G=[]  
    T=[]  
    for i in range(NC):  
        T.append(0)  
    for i in LJ:  
        G.append(0)  
    j=0  
    i=0  
    for Pi in LJ:  
        T[j] +=Pi  
        G[i]= j+1  
        i +=1  
  
        if T[j] >= Avg:  
  
            j=j+1  
  
    print 'Average Time for each crane is', Avg  
    print 'List of Total Time for each group is', T  
    print 'List of crane assigned to each job is', G  
    print 'makespan is', max(T)  
    return max(T)
```

```

def G2(LJ,NC):

    print 'Greedy 2'

    LJ.reverse()

    Total = sum(LJ)

    Avg = float(Total)/NC

    G=[]

    T=[]

    for i in range(NC):

        T.append(0)

        for i in LJ:

            G.append(0)

            j=0

            i=0

            for Pi in LJ:

                T[j] +=Pi

                G[i] = j+1

                i +=1

                if T[j] >= Avg:

                    j=j+1

            i=0

        for Pi in G :

            G[i]=(NC+1)-G[i]

            i+=1

    T.reverse()

    G.reverse()

    print 'Average Time for each crane is', Avg

    print 'List of Total Time for each group is', T

    print 'List of crane assigned to each job is', G

    print 'makespan is', max(T)

    return max(T)

```

```

def G3(LJ,NC):
    print 'Greedy 3'
    Total = sum(LJ)

    Avg = float(Total)/NC
    G=[]
    T=[]
    for i in range(NC):
        T.append(0)
    for i in LJ:
        G.append(0)
        j=0
        i=0
        for Pi in LJ:
            T[j] +=Pi
            G[i] = j+1
            i +=1
            if j != NC-1:
                if T[j] >= Avg:
                    T[j] -= Pi
                    j=j+1
                    T[j] += Pi
                    G[i-1]=j+1

    print 'Average Time for each crane is', Avg
    print 'List of Total Time for each group is', T
    print 'List of crane assigned to each job is', G
    print 'makespan is', max(T)
    return max(T)

```



```

def G4(LJ,NC):

    print 'Greedy 4'

    LJ.reverse()

    Total = sum(LJ)

    Avg = float(Total)/NC

    G=[]

    T=[]

    for i in range(NC):

        T.append(0)

        for i in LJ:

            G.append(0)

            j=0

            i=0

            for Pi in LJ:

                T[j] +=Pi

                G[i] =j+1

                i +=1

                if j != NC-1:

                    if T[j] >= Avg:

                        T[j] -= Pi

                        j=j+1

                        T[j] += Pi

                        G[i-1]=j+1

            i=0

        for Pi in G :

            G[i]=(NC+1)-G[i]

            i+=1

    T.reverse()

    G.reverse()

    print 'Average Time for each crane is', Avg

    print 'List of Total Time for each group is', T

    print 'List of crane assigned to each job is', G

```

```
print 'makespan is', max(T)  
return max(T)
```



```

def G5(LJ,NC):

    print 'Greedy 5'

    Total = sum(LJ)

    Avg = float(Total)/NC

    G=[]

    T=[]

    for i in range(NC):

        T.append(0)

    for i in LJ:

        G.append(0)

    j=0
    i=0

    for Pi in LJ:

        T[j] +=Pi
        G[i] = j+1
        i +=1
        if j != NC-1:

            if T[j] >= Avg:

                C1=T[j]-Avg
                C2=Avg-(T[j]-Pi)
                if C1<C2:

                    j=j+1

            else:

                T[j] -= Pi
                j=j+1
                T[j] += Pi
                G[i-1]=j+1

    print 'Average Time for each crane is', Avg
    print 'List of Total Time for each group is', T
    print 'List of crane assigned to each job is', G
    print 'makespan is', max(T)

    return max(T)

```

```

def G6(LJ,NC):
    print 'Greedy 6'
    LJ.reverse()
    Total = sum(LJ)
    Avg = float(Total)/NC
    G=[]
    T=[]
    for i in range(NC):
        T.append(0)
    for i in LJ:
        G.append(0)
    j=0
    i=0
    for Pi in LJ:
        T[j] +=Pi
        G[i] =j+1
        i +=1
        if j != NC-1:
            if T[j] >= Avg:
                C1=T[j]-Avg
                C2=Avg-(T[j]-Pi)
                if C1<C2:
                    j=j+1
            else:
                T[j] -= Pi
                j=j+1
                T[j] += Pi
                G[i-1]=j+1
    i=0
    for Pi in G :
        G[i]=(NC+1)-G[i]
        i+=1

```

```
T.reverse()
G.reverse()
print 'Average Time for each crane is', Avg
print 'List of Total Time for each group is', T
print 'List of crane assigned to each job is', G
print 'makespan is', max(T)
return max(T)
```



```

def G7(LJ,NC):
    print 'Greedy 7'
    NS=1000
    Sol=[]
    AllT=[]
    AllPL=[]
    for a in range(NS):
        PL=[]
        T = []
        Begin = 1
        Remain=NC-1
        End=len(LJ)-Remain
        for i in range(NC):
            PL.append(0)
            T.append(0)
            for i in range(NC):
                if i !=NC-1:
                    PL[i]=random.randint(Begin,End)
                    Begin = PL[i]+1
                    Remain -= 1
                    End = len(LJ)-Remain

            else:
                PL[i]=len(LJ)

        S = 0
        E = PL[0]
        for i in range(NC):
            T[i] = sum(LJ[S:E])
            S = E
            if i != NC-1:
                E = PL[i+1]

            else:

```

```
E=len(LJ)-1
AllT.append(T)
AllPL.append(PL)
Sol.append(max(T))
print 'The best solution is:', min(Sol)
print 'found at:', Sol.index(min(Sol))
print 'The solution is', AllT[Sol.index(min(Sol))]
print 'Each crane process as follows', AllPL[Sol.index(min(Sol))]
return min(Sol)
```



```
import random

import time

tt0=time.time()

L=[23,5,23,41,6,58,23,24] #Ex.processing time

EachSol=[]

for i in range(7):

    EachSol.append(0)

EachSol[0]= G1(L,3) #Ex.number of crane

EachSol[1]= G2(L,3) #Ex.number of crane

EachSol[2]= G3(L,3) #Ex.number of crane

EachSol[3]= G4(L,3) #Ex.number of crane

EachSol[4]= G5(L,3) #Ex.number of crane

EachSol[5]= G6(L,3) #Ex.number of crane

EachSol[6]= G7(L,3) #Ex.number of crane

print EachSol

print 'the best solution is', min(EachSol)

print 'for algorithm number', EachSol.index(min(EachSol)) + 1

ttAll=time.time()-tt0

print 'Running Time is', ttAll, 'Seconds'
```



## ประวัติผู้ดำเนินโครงการ



ชื่อ นายธนชัย เกตุเทียน  
 ภูมิลำเนา 71/4 ม.6 ต.สายลำโพง อ.ท่าตะโก  
 จ.นครสวรรค์

### ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนสามมิตร
- จบระดับมัธยมศึกษาจากโรงเรียนท่าตะโกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4  
 สาขาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-mail: [Thanachai\\_K@windowslive.com](mailto:Thanachai_K@windowslive.com)



ชื่อ นายพรศักดิ์ วัตสรวิงค์  
 ภูมิลำเนา 46/1 ม.4 ต.บึงปลาหู อ.บรรพตพิสัย  
 จ.นครสวรรค์

### ประวัติการศึกษา

- จบระดับประถมศึกษาจาก โรงเรียนราษฎร์ร่วม-  
สามัคคี
- จบระดับมัธยมศึกษาจาก โรงเรียนรัฐราษฎร์อนุสรณ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4  
 สาขาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-mail: [Ollysung@hotmail.com](mailto:Ollysung@hotmail.com)