

## CPE217 – Homework 7

### Homework: Stock Market Simulation using Priority Queues

Homework Due Date: 6 December 2016

Patiwet Wuttisarnwattana, Ph.D.

Department of Computer Engineering

- คำชี้แจงการส่งงาน
- ให้นักศึกษาส่งงานเข้าอีเมล [class.submissionx@gmail.com](mailto:class.submissionx@gmail.com) เท่านั้น ห้ามส่งงานเข้าเมลของอาจารย์ โปรดจำไว้ว่าอีเมลนี้อาจารย์จะไม่อ่านจนกว่าจะถึงเวลาตรวจ เพราะฉะนั้นถ้านักศึกษามีคำถามจะถามอาจารย์หรือพี่ TA ให้ส่งอีเมลเข้าอีเมลส่วนตัวอาจารย์หรือพี่ TA ตามที่ให้ไว้
- ในการส่งงานให้ระบุในช่อง Subject ของ email ให้เขียน pattern ดังต่อไปนี้ [CPE217][HWx][core person ID] ตรง x ให้ระบุว่าเป็นการบ้านที่เท่าไรและตรง core person ID ให้ระบุรหัสนักศึกษาที่เป็น core person ตัวอย่างเช่น [CPE217][HW7][580610615] หาก core person คือ กนกวรรณ
- แต่ละกลุ่ม ควรให้ core person เป็นคนส่งงาน และในเนื้อความอีเมล ให้ระบุ รหัสประจำตัวนักศึกษาของทุกคนที่เป็นสมาชิกในกลุ่ม หาก core person ไม่สามารถส่งงานได้ ให้สมาชิกคนอื่นก็ได้ส่งงานแทน แต่ core person id ต้องเป็นรหัสเดิมทุกครั้ง
- โค้ดของคุณต้องมีคอมเมนต์ (comment) เพื่ออธิบายว่าโค้ดดังที่เห็นอยู่นี้ทำงานอะไร หรือ if นี่ทำตรวจสอบอะไร หากกลุ่มไหนไม่มีคอมเมนต์ในโค้ดจะไม่ได้รับการตรวจ การเขียนคอมเมนต์ไม่ต้องเขียนแบบละเอียดยิบก็ได้ เท่าที่คุณต้องการให้ผู้ตรวจทราบก็พอ
- อาจารย์เปลี่ยนใจละ ไม่ต้องส่งคะแนน contribution ของเพื่อนในกลุ่ม หากกลุ่มไหนทนนไม่ไหวและอยากรายงานความประพฤติของเพื่อนว่าเขาเปรี้ยบ ไม่ช่วยทำงาน ให้ส่งอีเมลแจ้งอาจารย์เป็นกรณีไป อันที่จริงผลกรรมของการเขาเปรี้ยบจะไปแสดงตอนสอบไฟนอลอยู่ละ คุณสบายใจได้
- งานที่ส่งต้องประกอบด้วย Zip file ของ src folder ที่สามารถกด F6 รันได้เลย หากมี compile error หรือ runtime exception งานของนักศึกษาจะไม่ได้รับการตรวจ
- สามารถส่งการบ้านซ้ำได้ แต่หักคะแนนวันละ 10%

การบ้านนี้นักศึกษาจะได้เรียนรู้การทำงานของตลาดหลักทรัพย์ ซึ่งเป็นแหล่งแลกเปลี่ยน/ซื้อขายหุ้นกันระหว่างนักลงทุน เช่นนักลงทุน A มีหุ้นของบริษัท XYZ มีความต้องการใช้เงินด่วน จึงตัดสินใจส่งคำสั่งเสนอขาย (Offer) ไปยังระบบซื้อขายของตลาดหลักทรัพย์ นักลงทุน A จะขายไม่ได้จนกว่าจะมีคนส่งคำสั่งเสนอซื้อเข้ามา ต่อมาไม่นาน นักลงทุน B ซึ่งต้องการเข้าถือหุ้นบริษัท XYZ อยู่พอดี นักลงทุน B จึงส่งคำสั่งเสนอซื้อ (Bid) เข้ามาในระบบ ถ้าหากว่าราคาเสนอซื้อเท่ากับราคาเสนอขายพอดี การแลกเปลี่ยนซื้อขายก็จะเกิดขึ้น (Matched) โดยนักลงทุน A จะโอนหุ้นให้กับนักลงทุน B ส่วนนักลงทุน B ก็จะโอนเงินสดเข้าบัญชีของนักลงทุน A โดยระบบคอมพิวเตอร์จะทำหน้าที่นี้ให้โดยอัตโนมัติ

ระบบคอมพิวเตอร์ที่จะดูแลการจับคู่คำสั่งเสนอซื้อและเสนอขายนี้เรียกว่า Automatic Order Matching (AOM) ในความเป็นจริงแล้วในตลาดหลักทรัพย์จะมีนักลงทุนจำนวนมาก แต่ละคนก็จะเสนอซื้อและเสนอขายในราคาที่แตกต่างกัน ระบบ AOM จะมีวิธีการจัดการคำสั่งที่รับเข้ามาดังนี้

1. การจัดเรียงลำดับคำสั่งซื้อขายเมื่อสามารถส่งคำสั่งซื้อขายเข้ามา ระบบการซื้อขายจะเก็บคำสั่งซื้อขายไว้ตั้งแต่เวลาที่ส่งคำสั่งซื้อขาย จนถึงสิ้นวันทำการ และจัดเรียงคำสั่งซื้อขายตามลำดับของราคาและเวลาที่ดียที่สุด (Price then Time Priority) โดยมีหลักการคือ
  - 1.1. คำสั่งซื้อที่มีราคาเสนอซื้อสูงที่สุดจะถูกจัดเรียงไว้ในลำดับที่หนึ่ง และถ้ามีราคาเสนอซื้อที่สูงกว่าถูกส่งเข้ามาใหม่ จะจัดเรียง ราคาเสนอซื้อที่สูงกว่าเป็นการเสนอซื้อในลำดับแรกก่อนและถ้ามีการเสนอซื้อในแต่ละราคามากกว่าหนึ่งรายการ ให้จัดเรียงตามเวลา โดยการเสนอซื้อที่ปรากฏในระบบการซื้อขายก่อนจะถูกจัดไว้เป็นการเสนอซื้อในลำดับก่อน
  - 1.2. คำสั่งขายที่มีราคาเสนอขายต่ำที่สุดจะถูกจัดเรียงไว้ในลำดับที่หนึ่ง และถ้ามีราคาเสนอขายที่ต่ำกว่าถูกส่งเข้ามาใหม่จะจัดเรียงราคาเสนอขายที่ต่ำกว่า เป็นการเสนอขายในลำดับแรกก่อนและถ้ามีการเสนอขายในแต่ละราคามากกว่าหนึ่งรายการให้จัดเรียงตามเวลา โดยการเสนอขายที่ปรากฏในระบบการซื้อขาย ก่อนจะถูกจัดไว้เป็นการเสนอขายในลำดับก่อน
2. การจับคู่การซื้อขาย (Matching) เมื่อคำสั่งซื้อขายผ่านเข้ามาในระบบซื้อขายแล้ว ระบบซื้อขายจะตรวจสอบว่าคำสั่งนั้นสามารถจับคู่กับคำสั่ง ด้านตรงข้ามได้ทันทีหรือไม่ ถ้าคำสั่งนั้นสามารถจับคู่ได้ทันที ระบบก็จะทำการจับคู่ให้ แต่ถ้าคำสั่งนั้น ไม่สามารถจับคู่ได้ ระบบจะจัดเรียงคำสั่ง ซื้อขายนั้นตามหลักการ Price then Time Priority ตามที่กล่าวข้างต้นเพื่อรอการจับคู่คำสั่งต่อไป เงื่อนไขในการจับคู่กันได้ของคำสั่งก็คือ หากราคาเสนอซื้อที่ดีที่สุด (สูงที่สุด) มีค่ามากกว่าหรือเท่ากับ ราคาเสนอขายที่ดีที่สุด (ต่ำที่สุด) การจับคู่กันของคำสั่งก็จะเกิดขึ้น

ในการบ้านนี้อาจารย์ได้ทำการ implement ระบบตลาดหลักทรัพย์ไว้หมดแล้ว ทั้งการเพิ่มนักลงทุน, การเพิ่มหุ้น, การกระจายหุ้นให้นักลงทุนตอนเริ่มแรก, การส่งคำสั่งซื้อ/คำสั่งขาย, การจับคู่กันได้ของคำสั่ง, การโอนหุ้นและโอนเงินเมื่อเกิดการจับคู่, การแสดงข้อมูล Portfolio, การแสดงข้อมูล Quote, และอื่น ๆ ยกเว้นเหลือเพียงแต่ระบบ Priority Queues ที่นักศึกษาต้อง implement ด้วย Binary Heap (ใช้วิธีการของ Array-as-a-Complete-Binary-Tree) เองแค่นั้นเอง

การบ้านนี้อาจารย์จะไม่บังคับให้นักศึกษาใช้ template ของอาจารย์ ถ้าหากนักศึกษารอ่าน template ของอาจารย์แล้วไม่เข้าใจ นักศึกษาสามารถเขียนโค้ดใหม่ทั้งหมดอย่างไรก็ได้ ไม่ต้องใช้ Recursive ก็ได้ ขอเพียงแต่ นักศึกษาต้อง implement Priority Queue ด้วย Array-as-a-Complete-Binary-Tree และผลลัพธ์ต้องมี Output pattern เหมือนของอาจารย์เท่านั้นเอง

ในการบ้านนี้เราจะมี class StockMarket ทำหน้าที่ดำเนินการเป็นตลาดหลักทรัพย์ ที่จะบันทึกข้อมูลของนักลงทุน (class Investor), ข้อมูลของหุ้น (class Stock), ข้อมูล Priority Queues ที่จัดลำดับคำสั่งเสนอซื้อ/เสนอขายที่ดีที่สุดไว้ (class Heap), ระบบบัญชีที่บันทึกว่านักลงทุนแต่ละคนมีหุ้นแต่ละบริษัทกี่หุ้น (int[][] stockOwnership)

ต่อไปข้างล่างนี้คือการทำงานของโค้ด โดยอาจารย์ได้ implement ให้เกือบเสร็จแล้ว

1. สมมติว่าเราต้องการสร้าง Object ของ StockMarket โดยรองรับนักลงทุนได้ 10 คน มีหุ้นบริษัทอยู่ 10 บริษัท โค้ดของเราจะเขียนได้ว่า

```
StockMarket market = new StockMarket(10, 10);
```

2. สมมติว่าเรามีนักลงทุน 10 คนดังต่อไปนี้

รหัสนักลงทุน (Auto-generated)	ชื่อนักลงทุน	เงินทุนเริ่มแรก (บาท)
0	Darunphop	10000
1	Aunnop	10000
2	Pimukthee	10000
3	Uttakran	9000
4	Potsawat	9000
5	Suttawee	9000
6	Supakarn	8000
7	Tanin	8000
8	Theerapat	8000
9	Kawewut	8000

คำสั่งการเพิ่มนักลงทุนก็จะเขียนได้ว่า

```
market.addInvestor("Darunphop", 10000);
market.addInvestor("Aunnop", 10000);
market.addInvestor("Pimukthee", 10000);
market.addInvestor("Uttakran", 9000);
market.addInvestor("Potsawat", 9000);
market.addInvestor("Suttawee", 9000);
market.addInvestor("Supakarn", 8000);
market.addInvestor("Tanin", 8000);
market.addInvestor("Theerapat", 8000);
market.addInvestor("Kawewut", 8000);
```

### 3. สมมติว่าเรามีหุ้นอยู่ 10 บริษัท

รหัสบริษัท (Auto-generated)	รหัสย่อบริษัท	ราคาต่อหุ้นเริ่มแรก (บาท/หุ้น)
0	PTT	344
1	CPALL	60.50
2	SCB	144
3	KBANK	170
4	CPF	27.75
5	TRUE	7.15
6	CPN	53
7	BTS	8.35
8	DTAC	35.50
9	LH	8.75

คำสั่งการเพิ่มหุ้นก็จะเขียนได้ว่า

```
market.addStock("PTT", 344);
market.addStock("CPALL", 60.5);
market.addStock("SCB", 144);
market.addStock("KBANK", 170);
market.addStock("CPF", 27.75);
market.addStock("TRUE", 7.15);
market.addStock("CPN", 53);
market.addStock("BTS", 8.35);
market.addStock("DTAC", 35.50);
market.addStock("LH", 8.75);
```

- ขั้นตอนต่อไปเป็นการที่นักลงทุนจะเข้าซื้อหุ้นกับบริษัทโดยตรงในตลาดแรก หรือ Initial Public Offering (IPO) ในขั้นตอนนี้อาจารย์อยากให้นักศึกษาเข้าใจว่า หมายถึงการที่นักลงทุนเข้าซื้อหุ้นเข้าพอร์ตตัวเอง (Portfolio) โดยการซื้อนี้ทำโดยการสุ่ม อาจารย์ได้กำหนดเอาไว้ว่า ให้ใช้เงินเพียงแค่ครึ่งเดียวจากที่มีแต่แรกในการซื้อหุ้น เช่น Darunphop มีเงิน 10000 บาทแต่แรก ในขั้นนี้ Darunphop ก็จะสุ่มเข้าซื้อหุ้นต่าง ๆ ด้วยเงินประมาณครึ่งหนึ่ง การทำแบบนี้อาจารย์เขียนโค้ดไว้ในฟังก์ชันที่ชื่อว่า

```
market.simulateIPO();
```

- ขั้นตอนต่อไปเป็นการตรวจสอบว่า นักลงทุนแต่ละคนมีรายละเอียดบัญชีหุ้น หรือ พอร์ต (Portfolio) ตัวเองเป็นอย่างไร สมมติว่า Aunnop และ Darunphop อยากรู้ว่าตอนนี้พอร์ตตัวเองหน้าตาเป็นอย่างไร ก็จะพิมพ์คำสั่งดังต่อไปนี้

```
market.portfolio("Aunnop");  
market.portfolio("Darunphop");
```

ผลลัพธ์ก็จะแสดงว่า เหลือเงินในบัญชีที่จะซื้อหุ้นเพิ่มได้อีกกี่บาท และมีหุ้นของบริษัทต่าง อย่างละกี่หุ้น และมีมูลค่าหุ้นรวมเท่าไร

-----  
Portfolio of 'Aunnop', ID = 1  
Current balance is 2390.0 baht (Buying power)  
-----

Stock	Amount(shares)	Position Value(baht)
PTT	20	6880.0
CPALL	0	0.0
SCB	0	0.0
KBANK	0	0.0
CPF	20	555.0
TRUE	0	0.0
CPN	0	0.0
BTS	0	0.0
DTAC	0	0.0
LH	20	175.0

-----

Total portfolio values = 10000.0 bath  
-----

-----  
Portfolio of 'Darunphop', ID = 0  
Current balance is 4649.5 baht (Buying power)  
-----

Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	10	605.0
SCB	0	0.0
KBANK	0	0.0
CPF	10	277.5
TRUE	20	143.0
CPN	10	530.0
BTS	0	0.0
DTAC	10	355.0
LH	0	0.0

-----

Total portfolio values = 10000.0 bath  
-----

จากตัวอย่าง Aunnop ซึ่งมีเลขบัญชีคือ ID=1 มีเงินเหลือในบัญชีอีก 2390 บาท (เงินตรงนี้สามารถซื้อหุ้นเพิ่มได้อีก) และในพอร์ตแสดงรายละเอียดการถือหุ้น ซึ่งมีหุ้น PTT อยู่ 20 หุ้น มีหุ้น CPF อยู่ 20 หุ้นและมีหุ้น LH อีก 20 หุ้น มูลค่าสินทรัพย์ตอนนี้รวมอยู่ที่ 10000 บาท ส่วนพอร์ตของ Darunphop อาจารย์เข้าใจว่านักศึกษาสามารถอ่านเองได้ว่าหมายถึงอะไร

6. ขั้นตอนต่อไปเป็นการตรวจสอบทั้งตลาดเลยว่า นักลงทุนทั้งตลาดมีหุ้นอะไรบ้าง ให้ใช้คำสั่งดังต่อไปนี้

```
market.showStockOwnership();
```

ซึ่งจะให้ผลลัพธ์ดังต่อไปนี้

Number of shares each investor owns										
Inv.ID	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[PTT]	10	20	10	10	10	10	10	10	10	10
[CPALL]	10	0	10	0	20	0	0	10	0	0
[SCB]	0	0	10	0	0	0	10	0	0	0
[KBANK]	0	0	0	10	0	0	0	0	0	0
[CPF]	10	20	10	0	0	10	0	10	10	0
[TRUE]	20	0	10	0	0	0	10	0	10	10
[CPN]	10	0	0	10	0	10	0	0	0	20
[BTS]	0	0	0	10	0	20	10	0	0	0
[DTAC]	10	0	10	10	0	10	0	10	10	10
[LH]	0	20	0	0	10	0	0	10	0	0

ผลลัพธ์จะแสดงสรุปว่า นักลงทุนแต่ละคนที่แสดงด้วย ID ต่าง ๆ ตามแนวนอน มีหุ้นบริษัทต่างๆ ซึ่งอยู่แนวตั้ง อย่างละกี่หุ้น ยกตัวอย่างเช่น นักลงทุน ID [0] ซึ่งก็คือ Darunphop มีหุ้น PTT 10 หุ้น, มีหุ้น CPALL อยู่ 10 หุ้น, มีหุ้น CPF 10 หุ้น, มีหุ้น TRUE 20 หุ้น, มีหุ้น CPN และหุ้น DTAC อย่างละ 10 หุ้น ส่วนคนอื่น ๆ ก็ใช้หลักการอ่านเดียวกัน

7. นักศึกษาจะสังเกตเห็นว่า คำสั่ง market.simulateIPO(); จะสุ่มจำนวนหุ้นเข้าพอร์ตทุก ๆ ครั้งที่รันใหม่ (สังเกตด้วยการรันคำสั่ง market.showStockOwnership() ) เพื่อให้การบ้านนี้ง่ายต่อการคาดเดาผลลัพธ์ อาจารย์ได้สร้างอีกฟังก์ชันหนึ่งซึ่งจะกำหนดจำนวนหุ้นเริ่มแรกคงที่ทุกครั้งที่รันใหม่ ชื่อว่า

```
market.loadSavedIPO();
```

ผลลัพธ์จากการรันฟังก์ชันนี้ จะให้ผลลัพธ์เดียวกันกับข้อที่ 6. ที่แล้ว นักศึกษาสามารถใช้ ฟังก์ชัน market.loadSavedIPO() นี้แทนฟังก์ชัน market.simulateIPO() ได้เลย หรือจะใช้ market.simulateIPO() ตลอดการบ้านเพื่อเพิ่มความยากก็ได้ เอาที่ขอบเลย

8. ต่อไปเป็นการเสนอขาย [นักศึกษาต้องทำการ implement Heap เพื่อรองรับการทำงานตั้งแต่นี้เป็นต้นไป]

จากตารางข้อ 6 นักศึกษาสังเกตว่า จะมีนักลงทุนแค่สองคนเท่านั้น ที่ถือหุ้นของธนาคารไทยพาณิชย์ (SCB) คือ Pimukthee 10 หุ้น และ Supakarn 10 หุ้น ทั้งสองคนเห็นว่าตัวเองถือหุ้นของบริษัทชั้นดี มีการเติบโตในอนาคตสูง และเป็นที่ต้องการของนักลงทุนคนอื่น จึงจะตั้งราคาขายไว้สูง Pimukthee จึงส่งคำสั่งเสนอขายหุ้น SCB ไว้ที่หุ้นละ 200 บาท จำนวน 5 หุ้น (ไม่ยอมขายหมด) ส่วน Supakarn เสนอขายที่หุ้นละ 180 บาท จำนวน 10 หุ้น (ขายหมดเลย ถูกกว่าด้วย) ทั้งสอง (ซึ่งปกติจะไม่รู้จักกัน) จึงต่างคนต่างส่งคำสั่งไปที่ระบบ AOM ดังนี้

```
market.submitSellOrder("Pimukthee", "SCB", 5, 200);  
market.submitSellOrder("Supakarn", "SCB", 10, 180);
```

9. นักลงทุนคนอื่น ๆ ที่สนใจในหุ้น SCB หากต้องการใช้ราคาเสนอขายปัจจุบันของ SCB ค่าเหล่านั้นก็สามารถที่จะตรวจสอบ “การเสนอราคา” โดยคำสั่งดังต่อไปนี้

```
market.quote("SCB");
```

ผลลัพธ์ก็จะแสดงว่า

```
Stock 'SCB':      [Best Bid = 0x0.0 baht,      Best Offer = 10x180.0 baht]
```

หมายความว่า ผู้ที่เสนอราคาขายหุ้น SCB ที่ดีที่สุด (Best Offer) อยู่ที่ 180 บาท 10 หุ้น ส่วนตอนนี้ยังไม่มีคนเสนอราคาซื้อ (Best Bid) = 0 ปกติระบบจะไม่แสดงให้คนอื่นเห็นว่าตอนนี้ ใครเป็นผู้เสนอราคาที่ดีที่สุดอยู่

ทั้งนี้หากมีคนเสนอราคาที่ดีที่สุดเท่ากันหลาย ๆ คน ระบบจะให้คนที่มาก่อนเท่านั้นที่จะเป็นคนอยู่ด้านบนสุดของ Heap ตามหลักการของ Priority Queue

10. ต่อมานักลงทุน Uttakran และ Potsawat ซึ่งตอนนี้ยังไม่มีหุ้น SCB เห็นศักยภาพของบริษัท SCB ต้องการจะเป็นเจ้าของ SCB (เป็นผู้ถือหุ้น) อยากได้หุ้น SCB มาถือกันคนละ 6 หุ้น จึงจะเสนอซื้อหุ้น SCB ผู้ราคาได้เต็มที่ไม่เกิน 250 บาทต่อหุ้น ทั้งสองจึงส่งคำสั่งดังนี้

```
market.submitBuyOrder("Uttakran", "SCB", 6, 250);  
market.submitBuyOrder("Potsawat", "SCB", 6, 250);
```

ผลลัพธ์จะแสดงดังนี้

```
-----  
Matched!!! Now 6 shares of Stock SCB are transferred from 'Supakarn' to 'Uttakran'  
Also, 1080.0 baht is transferred from 'Uttakran' to 'Supakarn'  
-----
```

```
-----  
Matched!!! Now 4 shares of Stock SCB are transferred from 'Supakarn' to 'Potsawat'  
Also, 720.0 baht is transferred from 'Potsawat' to 'Supakarn'  
-----
```

```
-----  
Matched!!! Now 2 shares of Stock SCB are transferred from 'Pimukthee' to 'Potsawat'  
Also, 400.0 baht is transferred from 'Potsawat' to 'Pimukthee'  
-----
```

จะเห็นว่า ราคาเสนอซื้อที่ทั้งสองคนส่งเข้าระบบ Match กับราคาที่เสนอขายของทั้ง Supakarn และ Pimukthee ผลก็คือหุ้น SCB 6 หุ้นของ Supakarn จึงแลกเปลี่ยนกับเงินสดของ Uttakran โดยมูลค่าเงินที่ Uttakran ต้องส่งเข้าบัญชีของ

Supakarn จะเท่ากับ จำนวนหุ้น x ราคาที่ตกลงกันได้ (Deal price) =  $6 \times 180 = 1080$  บาท หลังจากนั้นคำสั่งเสนอขายของ Supakarn จึงต้องถูกปรับปรุงจากเสนอขาย 10 หุ้นเหลือ เสนอขาย 4 หุ้น

ส่วนคำสั่งซื้อ 6 หุ้นของ Potsawat ซึ่งส่งคำสั่งเข้ามาช้ากว่า จึง Match กับ 4 หุ้นที่เหลือของ Supakarn (Deal price ในกรณียังคงเป็น 180 ตามที่ Supakarn ตั้งราคาขายไว้) ส่วนอีก 2 หุ้นที่เหลือจึงเข้าไป Match กับคิวถัดไปนั่นก็คือ Pimukthee เพราะคำสั่งซื้อของ Supakarn เสนอซื้อสูงกว่าราคาเสนอขายของ Pimukthee ที่ 200 บาทต่อหุ้น ด้วยเหตุนี้ deal price จึงเป็น 200 บาท สรุป Potsawat ต้องส่งเงินจำนวน  $4 \times 180 = 720$  เข้าบัญชี Supakarn และต้องส่งเงิน  $2 \times 200 = 400$  บาทเข้าบัญชี Pimuthee

สังเกตว่า คำสั่งเสนอขายของ Pimukthee ยังไม่สิ้นสุด ยังเหลือ เสนอขายอีก 3 หุ้นสุดท้าย หากมีคนต้องการซื้อ จะสามารถซื้อได้ 3 หุ้นนี้เท่านั้น ที่ราคา 200 บาท ยกเว้นแต่ถ้าจะมีใครส่งคำสั่งเสนอขายเข้ามาอีก หรือถ้าเสนอขายที่ราคา ต่ำกว่านี้ เช่น 150 บาทต่อหุ้น คิวการเสนอขายของ Pimukthee ก็จะถูกผลักลึกลงไป Heap โดยทันที

ถ้าหากเราต้องการตรวจสอบบัญชีปัจจุบันของตัวละครทั้งสี่คน เราสามารถใช้คำสั่งนี้

```
market.portfolio("Uttakran");
market.portfolio("Potsawat");
market.portfolio("Pimukthee");
market.portfolio("Supakarn");
```

ซึ่งจะให้ผลลัพธ์ดังนี้

-----		
Portfolio of 'Uttakran', ID = 3		
Current balance is 1811.5 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	0	0.0
SCB	6	1200.0
KBANK	10	1700.0
CPF	0	0.0
TRUE	0	0.0
CPN	10	530.0
BTS	10	83.5
DTAC	10	355.0
LH	0	0.0
-----		
Total portfolio values = 9120.0 bath		
-----		
-----		
Portfolio of 'Potsawat', ID = 4		
Current balance is 3142.5 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	20	1210.0
SCB	6	1200.0
KBANK	0	0.0



CPF	0	0.0
TRUE	0	0.0
CPN	0	0.0
BTS	0	0.0
DTAC	0	0.0
LH	10	87.5
-----		
Total portfolio values = 9080.0 bath		
-----		
-----		
Portfolio of 'Pimukthee', ID = 2		
Current balance is 4211.0 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	10	605.0
SCB	8	1600.0
KBANK	0	0.0
CPF	10	277.5
TRUE	10	71.5
CPN	0	0.0
BTS	0	0.0
DTAC	10	355.0
LH	0	0.0
-----		
Total portfolio values = 10560.0 bath		
-----		
-----		
Portfolio of 'Supakarn', ID = 6		
Current balance is 4765.0 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	0	0.0
SCB	0	0.0
KBANK	0	0.0
CPF	0	0.0
TRUE	10	71.5
CPN	0	0.0
BTS	10	83.5
DTAC	0	0.0
LH	0	0.0
-----		
Total portfolio values = 8360.0 bath		
-----		

จะเห็นได้ว่า เงินสดของ Uttakran และ Potsawat ลดลงเมื่อเทียบกับเงินต้นแต่แรก แต่ทั้งสองคนนี้ก็ยังมีหุ้น SCB เข้ามาอยู่ในพอร์ตกันคนละ 6 หุ้น ส่วน Pitmukthee และ Supakarn มีหุ้น SCB หายไปคนละ 6 หุ้น แต่ก็ก็มีเงินสดเพิ่มขึ้นจากการขายหุ้น

11. ต่อไปเป็นการตั้งราคาเสนอซื้อให้มีหลาย ๆ คนบ้าง จากตารางข้อ 6 นักศึกษาจะสังเกตว่าหุ้นของธนาคารกรุงไทย (KBANK) ขายตลาดมาก ๆ ทุกคนในตลาดอยากได้หุ้นบริษัทนี้ แต่ไม่ยอมซื้อที่ราคาสูง เพราะจากการประเมินงบการเงินอะไรต่าง ๆ แล้ว หุ้น KBANK ควรจะมีราคาอยู่ประมาณ 100 บาทเท่านั้น Suttawee เป็นคนแรก que ที่เข้ามาซื้อราคาในตลาด ด้วยคำสั่ง market.quote("KBANK");

```
Stock 'KBANK': [Best Bid = 0x0.0 baht, Best Offer = 0x170.0 baht]
```

ข้อมูลแสดงว่า ยังไม่มีใครเสนอราคาซื้อและราคาขายแต่อย่างใด (สังเกต 0x ของทั้งสองฝั่ง) Suttawee จึงเป็นผู้กำหนดราคาคนแรกของตลาดเลย จึงส่งคำสั่งเสนอซื้อหุ้น KBANK จำนวน 2 หุ้น ที่ราคา 90 บาท (เพื่อว่าจะซื้อได้ถูกกว่าราคาประเมิน)

```
market.submitBuyOrder("Suttawee", "KBANK", 2, 90);  
market.quote("KBANK");
```

ผลลัพธ์

```
Stock 'KBANK': [Best Bid = 2x90.0 baht, Best Offer = 0x170.0 baht]
```

ต่อมานักลงทุน Supakarn, Tanin, Theerapat, Kawewut ต่างก็อยากถือหุ้น KBANK จึงเข้ามาเสนอซื้อที่ราคาต่าง ๆ ดังนี้

```
market.submitBuyOrder("Supakarn", "KBANK", 1, 85);  
market.quote("KBANK");  
market.submitBuyOrder("Tanin", "KBANK", 3, 90);  
market.quote("KBANK");  
market.submitBuyOrder("Theerapat", "KBANK", 2, 100);  
market.quote("KBANK");  
market.submitBuyOrder("Kawewut", "KBANK", 2, 80);  
market.quote("KBANK");
```

ผลลัพธ์ก็จะแสดงดังนี้

```
Stock 'KBANK': [Best Bid = 2x90.0 baht, Best Offer = 0x170.0 baht]  
Stock 'KBANK': [Best Bid = 2x90.0 baht, Best Offer = 0x170.0 baht]  
Stock 'KBANK': [Best Bid = 2x100.0 baht, Best Offer = 0x170.0 baht]  
Stock 'KBANK': [Best Bid = 2x100.0 baht, Best Offer = 0x170.0 baht]
```

12. เรื่องนี้น่าสนใจตรงที่ Uttakran เป็นเพียงคนเดียวที่มีหุ้น KBANK อยู่ในมือ, Uttakran ตระหนักดีว่า ตัวเองซื้อหุ้นนี้มาตอนที่ราคา 170 บาท บัดนี้ตลาดเสนอราคาซื้อ (ที่ดีที่สุด) แค่ 100 บาท/หุ้น Uttakran มีทางเลือกที่จะไม่ขายหุ้นนี้ก็ได้ (ถือกินเงินปันผล รอบริษัทโตไปเรื่อย ๆ ) หรือ รอให้คนที่เสนอซื้อให้ราคาสูงกว่านี้ในราคาที่ Uttakran รับผิดชอบแล้วค่อยขาย หรือ Uttakran อาจจะเสนอขายเองที่ราคาที่สูงกว่านี้ก็ได้ เช่น

```
market.submitSellOrder("Uttakran", "KBANK", 2, 150);  
market.quote("KBANK");
```

ผลลัพธ์ก็จะแสดงว่า

Stock 'KBANK': [Best Bid = 2x100.0 baht, Best Offer = 2x150.0 baht]

ซึ่งมันอาจจะเป็นอย่างนี้ตราบนานเท่านานก็ได้ ตราบใดที่ยังไม่มีใครเสนอราคาซื้อที่สูงกว่า 150 บาท

สมมุติเราเพิ่มเติมอีกหน่อยก็แล้วกัน ให้ Uttakran อยู่ในสถานะร้อนเงิน หุ้น KBANK เป็นหุ้นที่ไม่ดี Uttakran จึงตัดสินใจขาย แต่ขายแค่ 6 หุ้น และจะขายในราคาที่ไม่ต่ำกว่า 85 บาทต่อหุ้น Uttakran จึงส่งคำสั่งดังต่อไปนี้

```
market.submitSellOrder("Uttakran", "KBANK", 6, 85);
```

ผลลัพธ์จึงเป็นดังนี้

-----  
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Uttakran' to 'Theerapat'

Also, 200.0 baht is transferred from 'Theerapat' to 'Uttakran'

-----  
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Uttakran' to 'Suttawee'

Also, 180.0 baht is transferred from 'Suttawee' to 'Uttakran'

-----  
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Uttakran' to 'Tanin'

Also, 180.0 baht is transferred from 'Tanin' to 'Uttakran'

-----  
เนื่องจากมีผู้มารอซื้อจำนวนมาก จึงเกิดคำสั่ง Match ขึ้นมากมาย ในกรณีนี้ Uttakran เสนอขายในราคาค่อนข้างต่ำคือ 85 บาทต่อหุ้น จึงเกิดการ Match ทั้งหมด (แต่คนที่มารอซื้อไม่ได้ทั้งหมดเพราะ หุ้นที่เสนอขายมีน้อย) Theerapat ได้เสนอซื้อในราคาที่สูงที่สุด คือ 100 บาทต่อหุ้น จึงได้สิทธิ์แลกเปลี่ยน 2 หุ้นที่เสนอซื้อเข้ามาก่อน ได้ไปก่อน จึงโอนเงินไปให้  $2 \times 100 = 200$  บาท, คนต่อมาก็คือ Suttawee ซึ่งเสนอราคาซื้อไว้ที่ 90 บาทต่อหุ้น (และเสนอก่อนคู่แข่งคือ Tanin) จึงได้ไป 2 หุ้น โดยแลกกับเงิน  $2 \times 90 = 180$  บาท ส่วน Tanin เสนอซื้ออยู่ที่ 90 บาทเหมือนกัน แต่ช้ากว่า จึงได้ที่หลัง ถึงแม้ว่า Tanin จะเสนอซื้อ 3 หุ้นแต่ก็ได้ไปแค่ 2 หุ้นเพราะ Uttakran เสนอขายแค่ 6 หุ้น (ให้ไปกับคนก่อนหน้าแล้ว 4 จึงเหลือขายให้ Tanin แค่ 2 หุ้น) เพราะฉะนั้นคำสั่งที่ค้างเหลืออยู่ในระบบตอนนี้จึงเป็น คำสั่งซื้อของ Tanin 1 หุ้นที่ราคา 90 บาท, คำสั่งซื้อของ Kawewut 2 หุ้นที่ราคา 80 บาท และสุดท้ายคำสั่งขายของ Uttakran เอง 2 หุ้นแต่แรก ที่เสนอขายไว้ 150 บาท ถึงป่านี้อังยังไม่มีใครจะซื้อที่ราคานี้ หากตรวจสอบ Portfolio ของ Uttakran จะพบว่ามียาละเอียดดังนี้

```
market.portfolio("Uttakran");
```

-----  
Portfolio of 'Uttakran', ID = 3

Current balance is 2371.5 baht (Buying power)

Stock	Amount(shares)	Position Value(baht)
PTT	10	3440.0
CPALL	0	0.0
SCB	6	1200.0
KBANK	4	340.0
CPF	0	0.0
TRUE	0	0.0
CPN	10	530.0
BTS	10	83.5
DTAC	10	355.0
LH	0	0.0
-----		
Total portfolio values = 8320.0 bath		
-----		

จะเห็นได้ว่า Uttakran มีหุ้น KBANK ลดลงจาก 10 หุ้นเหลือ 4 หุ้น (เพราะขายไปเมื่อที่ 6 หุ้น) สังเกตว่าสินทรัพย์โดยรวมมีมูลค่าลดลง เนื่องจากราคาหุ้นตก จากที่ซื้อมา 170 บาทต่อหุ้น ตอนนี้เหลือราคาที่ซื้อขายกันที่ 85 บาทต่อหุ้น เงินจึงหายวับจากการซื้อขายไปที่  $9000 - 8320 = 680$  บาท Uttakran ก็จะไปบอกกับเพื่อนๆว่า แค่นี้สิวะ ๆ

13. ต่อไปนี้เป็นการจำลองสถานการณ์ ที่จะให้นักลงทุน ทำการเสนอซื้อ/เสนอขายหุ้น PTT ที่ราคาต่าง ๆ โดยกำหนดให้จำนวนหุ้นเริ่มต้นเป็นไปตามข้อมูลนี้

```
market.showStockOwnership();
```

เริ่มต้นมีการกระจายหุ้นเป็นดังนี้

Number of shares each investor owns										
Inv.ID	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[PTT]	10	20	10	10	10	10	10	10	10	10
[CPALL]	10	0	10	0	20	0	0	10	0	0
[SCB]	0	0	8	6	6	0	0	0	0	0
[KBANK]	0	0	0	4	0	2	0	2	2	0
[CPF]	10	20	10	0	0	10	0	10	10	0
[TRUE]	20	0	10	0	0	0	10	0	10	10
[CPN]	10	0	0	10	0	10	0	0	0	20
[BTS]	0	0	0	10	0	20	10	0	0	0
[DTAC]	10	0	10	10	0	10	0	10	10	10
[LH]	0	20	0	0	10	0	0	10	0	0

การส่งคำสั่งเป็นดังต่อไปนี้

```
market.submitSellOrder("Darunphop", "PTT", 3, 350);  
market.submitSellOrder("Darunphop", "PTT", 3, 360);  
market.submitSellOrder("Darunphop", "PTT", 4, 370);  
market.submitSellOrder("Pimukthee", "PTT", 5, 355);  
market.submitSellOrder("Pimukthee", "PTT", 5, 365);  
market.submitSellOrder("Pimukthee", "PTT", 5, 375);  
market.submitSellOrder("Potsawat", "PTT", 3, 360);  
market.submitSellOrder("Potsawat", "PTT", 3, 370);  
market.submitSellOrder("Potsawat", "PTT", 4, 380);
```

```
market.submitBuyOrder("Aunnop", "PTT", 6, 370);  
market.submitBuyOrder("Uttakran", "PTT", 6, 370);  
market.submitBuyOrder("Suttawee", "PTT", 10, 370);  
market.submitBuyOrder("Supakarn", "PTT", 10, 370);
```

```
market.submitBuyOrder("Tanin", "PTT", 2, 330);  
market.submitBuyOrder("Tanin", "PTT", 2, 300);  
market.submitBuyOrder("Tanin", "PTT", 2, 270);
```

```
market.submitBuyOrder("Theerapat", "PTT", 3, 330);  
market.submitBuyOrder("Theerapat", "PTT", 3, 250);  
market.submitBuyOrder("Theerapat", "PTT", 3, 200);
```

```
market.submitSellOrder("Kawewut", "PTT", 3, 320);  
market.submitSellOrder("Kawewut", "PTT", 3, 240);  
market.submitSellOrder("Kawewut", "PTT", 4, 220);  
market.submitSellOrder("Aunnop", "PTT", 3, 320);  
market.submitSellOrder("Aunnop", "PTT", 3, 240);  
market.submitSellOrder("Aunnop", "PTT", 4, 220);
```

```
market.quote("PTT");
```

-----  
Matched!!! Now 3 shares of Stock PTT are transferred from 'Darunphop' to 'Aunnop'  
Also, 1050.0 baht is transferred from 'Aunnop' to 'Darunphop'  
-----  
-----

Matched!!! Now 3 shares of Stock PTT are transferred from 'Pimukthee' to 'Aunnop'  
Also, 1065.0 baht is transferred from 'Aunnop' to 'Pimukthee'  
-----  
-----

Matched!!! Now 2 shares of Stock PTT are transferred from 'Pimukthee' to 'Uttakran'  
Also, 710.0 baht is transferred from 'Uttakran' to 'Pimukthee'  
-----  
-----

Matched!!! Now 3 shares of Stock PTT are transferred from 'Darunphop' to 'Uttakran'  
Also, 1080.0 baht is transferred from 'Uttakran' to 'Darunphop'  
-----  
-----

Matched!!! Now 1 shares of Stock PTT are transferred from 'Potsawat' to 'Uttakran'  
Also, 360.0 baht is transferred from 'Uttakran' to 'Potsawat'  
-----  
-----

Matched!!! Now 2 shares of Stock PTT are transferred from 'Potsawat' to 'Suttawee'  
Also, 720.0 baht is transferred from 'Suttawee' to 'Potsawat'  
-----  
-----

Matched!!! Now 5 shares of Stock PTT are transferred from 'Pimukthee' to 'Suttawee'  
Also, 1825.0 baht is transferred from 'Suttawee' to 'Pimukthee'  
-----  
-----

Matched!!! Now 3 shares of Stock PTT are transferred from 'Darunphop' to 'Suttawee'  
Also, 1110.0 baht is transferred from 'Suttawee' to 'Darunphop'  
-----  
-----

Matched!!! Now 1 shares of Stock PTT are transferred from 'Darunphop' to 'Supakarn'

Also, 370.0 baht is transferred from 'Supakarn' to 'Darunphop'

---

---

Matched!!! Now 3 shares of Stock PTT are transferred from 'Potsawat' to 'Supakarn'

Also, 1110.0 baht is transferred from 'Supakarn' to 'Potsawat'

---

---

Matched!!! Now 3 shares of Stock PTT are transferred from 'Kawewut' to 'Supakarn'

Also, 1110.0 baht is transferred from 'Supakarn' to 'Kawewut'

---

---

Matched!!! Now 3 shares of Stock PTT are transferred from 'Kawewut' to 'Supakarn'

Also, 1110.0 baht is transferred from 'Supakarn' to 'Kawewut'

---

---

Matched!!! Now 2 shares of Stock PTT are transferred from 'Kawewut' to 'Tanin'

Also, 660.0 baht is transferred from 'Tanin' to 'Kawewut'

---

---

Matched!!! Now 2 shares of Stock PTT are transferred from 'Kawewut' to 'Theerapat'

Also, 660.0 baht is transferred from 'Theerapat' to 'Kawewut'

---

---

Matched!!! Now 1 shares of Stock PTT are transferred from 'Aunnop' to 'Theerapat'

Also, 330.0 baht is transferred from 'Theerapat' to 'Aunnop'

---

---

Matched!!! Now 2 shares of Stock PTT are transferred from 'Aunnop' to 'Tanin'

Also, 600.0 baht is transferred from 'Tanin' to 'Aunnop'

---

---

Matched!!! Now 1 shares of Stock PTT are transferred from 'Aunnop' to 'Tanin'

Also, 270.0 baht is transferred from 'Tanin' to 'Aunnop'

-----  
-----  
Matched!!! Now 1 shares of Stock PTT are transferred from 'Aunnop' to 'Tanin'

Also, 270.0 baht is transferred from 'Tanin' to 'Aunnop'

-----  
-----

Matched!!! Now 3 shares of Stock PTT are transferred from 'Aunnop' to 'Theerapat'

Also, 750.0 baht is transferred from 'Theerapat' to 'Aunnop'

-----

Stock 'PTT': [Best Bid = 3x200.0 baht, Best Offer = 2x320.0 baht]

เมื่อตรวจสอบการกระจายหุ้น

market.showStockOwnership();

Number of shares each investor owns										
Inv.ID	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[PTT]	0	18	0	16	4	20	20	16	16	0
[CPALL]	10	0	10	0	20	0	0	10	0	0
[SCB]	0	0	8	6	6	0	0	0	0	0
[KBANK]	0	0	0	4	0	2	0	2	2	0
[CPF]	10	20	10	0	0	10	0	10	10	0
[TRUE]	20	0	10	0	0	0	10	0	10	10
[CPN]	10	0	0	10	0	10	0	0	0	20
[BTS]	0	0	0	10	0	20	10	0	0	0
[DTAC]	10	0	10	10	0	10	0	10	10	10
[LH]	0	20	0	0	10	0	0	10	0	0

นักศึกษาสามารถตรวจสอบย้อนกลับไปได้หรือไม่ว่าทำไม ผลลัพธ์จึงออกมาเป็นแบบนี้



หากตรวจสอบ Portfolio ของนักลงทุนบางคน เช่น

market.portfolio("Aunnop");		
-----		
Portfolio of 'Aunnop', ID = 1		
Current balance is 2495.0 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	18	3960.0
CPALL	0	0.0
SCB	0	0.0
KBANK	0	0.0
CPF	20	555.0
TRUE	0	0.0
CPN	0	0.0
BTS	0	0.0
DTAC	0	0.0
LH	20	175.0
-----		
Total portfolio values = 7185.0 bath		
-----		

จะพบว่า Total Portfolio Values ของ Aunnop ได้หายไป  $10000 - 7185 = 2815$  บาท นักศึกษาสามารถตอบได้หรือไม่ว่าทำไมหายไปไหน?

14. สมมติว่าจากเหตุการณ์ข้อ 13 มีการซื้อขายกับอีกเล็กน้อยดังนี้

```
market.submitBuyOrder("Darunphop", "PTT", 20, 150);
market.submitBuyOrder("Darunphop", "PTT", 20, 100);
market.submitBuyOrder("Darunphop", "PTT", 10, 50);

market.submitSellOrder("Suttawee", "PTT", 10, 125);
market.submitSellOrder("Supakarn", "PTT", 10, 75);
market.submitSellOrder("Tanin", "PTT", 15, 75);
market.submitSellOrder("Theerapat", "PTT", 15, 75);

market.quote("PTT");
```

ให้ผลลัพธ์คือ

-----  
Matched!!! Now 3 shares of Stock PTT are transferred from 'Suttawee' to 'Theerapat'

Also, 600.0 baht is transferred from 'Theerapat' to 'Suttawee'

-----  
Matched!!! Now 7 shares of Stock PTT are transferred from 'Suttawee' to 'Darunphop'

Also, 1050.0 baht is transferred from 'Darunphop' to 'Suttawee'

-----  
Matched!!! Now 10 shares of Stock PTT are transferred from 'Supakarn' to 'Darunphop'

Also, 1500.0 baht is transferred from 'Darunphop' to 'Supakarn'

-----  
Matched!!! Now 3 shares of Stock PTT are transferred from 'Tanin' to 'Darunphop'

Also, 450.0 baht is transferred from 'Darunphop' to 'Tanin'

-----  
Matched!!! Now 12 shares of Stock PTT are transferred from 'Tanin' to 'Darunphop'

Also, 1200.0 baht is transferred from 'Darunphop' to 'Tanin'

-----  
Matched!!! Now 8 shares of Stock PTT are transferred from 'Theerapat' to 'Darunphop'

Also, 800.0 baht is transferred from 'Darunphop' to 'Theerapat'

-----  
Stock 'PTT': [Best Bid = 10x50.0 baht, Best Offer = 7x75.0 baht]

คุณสังเกตได้หรือไม่ว่า Aunnop ไม่ได้เกี่ยวข้องกับการซื้อขายกันในข้อ 14. เลย คราวนี้มาดู Portfolio ของ Aunnop กัน

market.portfolio("Aunnop");		
-----		
Portfolio of 'Aunnop', ID = 1		
Current balance is 2495.0 baht (Buying power)		
-----		
Stock	Amount(shares)	Position Value(baht)
PTT	18	1350.0
CPALL	0	0.0
SCB	0	0.0
KBANK	0	0.0
CPF	20	555.0
TRUE	0	0.0
CPN	0	0.0
BTS	0	0.0
DTAC	0	0.0
LH	20	175.0
-----		
Total portfolio values = 4575.0 bath		
-----		

คราวนี้มูลค่าบัญชีหุ้นของ Aunnop ลดลงอีก เหลือ 4575 เงินหายไป  $10000 - 4575 = 5425$  หรือ  $-54\%$  นักศึกษาทราบหรือไม่ว่า เงิน Aunnop หายไปไหน ทั้ง ๆ ที่ไม่ได้เกี่ยวข้องกับการซื้อขายหุ้นเลย ผมจะไปถามอีกทีตอนสอบครับ

- การบ้านนี้ผม implement ให้เกือบเสร็จหมดแล้วครับ เหลือเพียงแค่สองคลาสสุดท้ายเท่านั้น คือ **class Node** และ **class Heap** ที่คุณต้องแก้ไขโค้ดเพิ่มเติมจนเสร็จ และให้ทำงานได้ดั่งใจത്യด้านบน
- หากนักศึกษาอ่านโค้ดอาจารย์ไม่รู้เรื่องหรือไม่ชอบโค้ดอาจารย์ อาจารย์อนุญาตให้นักศึกษาเขียนเองได้ใหม่ทั้งหมด โดยไม่ต้องใช้ starter code ของอาจารย์ แต่ output pattern ต้องเหมือนเดิม และ Priority Queues ต้องถูก implement ด้วย Binary Heap
- หากการบ้านนี้นักศึกษาไม่รู้จะไปยังไง ไม่เข้าใจเลย หรือเสียเวลาการทำงานบ้านมากเกินไป ขอให้นักศึกษาเข้ามาปรึกษากับอาจารย์เป็นการด่วน เพื่อที่จะไม่เสียเวลาในการอ่านเตรียมสอบวิชาอื่นต่อไป