

CPE217 – Homework 8

Homework: Stock Market Simulation using Priority Queues

Homework Due Date: 16 November 2019

Patiwet Wuttisarnwattana, Ph.D.

Department of Computer Engineering

- คำชี้แจงการส่งงาน
- แต่ละกลุ่ม ควรให้ core person เป็นคนส่งงาน และในช่องข้อความต้องระบุรหัสประจำตัวนักศึกษาของทุกคนที่เป็นสมาชิกในกลุ่ม หาก core person ไม่สามารถส่งงานได้ ให้สมาชิกคนใดก็ได้ส่งงานแทน แต่ต้องบอกว่าส่งแทน core person ซึ่งก็คือใคร มีรหัสอะไร
- โค้ดของคุณต้องมีคอมเมนต์ (comment) เพื่ออธิบายว่าโค้ดดังที่เห็นอยู่นี้ทำงานอะไร หรือ if นี่ทำตรวจสอบอะไร หากกลุ่มไหนไม่มีคอมเมนต์ในโค้ดจะไม่ได้รับการตรวจ การเขียนคอมเมนต์ไม่ต้องเขียนแบบละเอียดยิบก็ได้เท่าที่คุณต้องการให้ผู้ตรวจทราบก็พอ
- งานที่ส่งต้องประกอบด้วย ZIP file ของ src folder ที่สามารถกด F6 รันได้เลย หากมี compile error หรือ runtime exception งานของนักศึกษาจะไม่ได้รับการตรวจ
- สามารถส่งการบ้านช้าได้ แต่หักคะแนนวันละ 10%
- การลอกงานเพื่อนมาส่ง เป็นการทุจริตและมีความผิดทางวินัย หากตรวจพบอาจารย์อาจพิจารณาให้คะแนนการบ้านนั้นหรือคะแนนการบ้านทั้งหมดของคุณ และ/หรือ คะแนนจิตพิสัย ทั้งหมดได้ศูนย์คะแนน ซึ่งนั่นอาจเป็นปัจจัยของการตัดสินใจถอนกระบวนวิชานี้ของคุณและลงทะเบียนใหม่ในปีการศึกษาหน้า
- การลอกงานมาส่งต้องรับผิดชอบพร้อมกันทั้งกลุ่ม จะให้คนทำผิด รับผิดชอบแต่เพียงคนเดียวไม่ได้
- เพื่อนในกลุ่มที่เหลือนี้อาจมีหน้าที่ต้องตรวจสอบความถูกต้องและรับประกันผลงานว่าไม่นำผลงานของกลุ่มอื่นมาส่ง

การบ้านนี้นักศึกษาจะได้เรียนรู้การทำงานของตลาดหลักทรัพย์ ซึ่งเป็นแหล่งแลกเปลี่ยนซื้อขายหุ้นกันระหว่างนักลงทุน เช่นนักลงทุน A มีหุ้นของบริษัท XYZ มีความต้องการใช้เงินสด 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 จะมีวิธีการจัดการคำสั่งที่รับเข้ามามีดังนี้

1. การจัดเรียงลำดับคำสั่งซื้อขายเมื่อสามารถส่งคำสั่งซื้อขายเข้ามา ระบบการซื้อขายจะเก็บคำสั่งซื้อขายไว้ตั้งแต่เวลาที่ส่งคำสั่งซื้อขาย จนถึงสิ้นวันทำการ และจัดเรียงคำสั่งซื้อขายตามลำดับของราคาและเวลาที่ดียิ่งที่สุด (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 ถ้าหากนักศึกษารับ 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), ระบบบัญชีที่บันทึกว่านักลงทุนแต่ละคนมีหุ้นแต่ละบริษัทที่หุ้น (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	Thanat	10000
1	Ronnachai	10000
2	Peerawich	10000
3	Apichart	9000
4	Piampong	9000
5	Phitchayarat	9000
6	Goravit	8000
7	Kittinan	8000
8	Waritthon	8000
9	Parinya	8000

คำสั่งการเพิ่มนักลงทุนก็จะเขียนได้ว่า

```
market.addInvestor("Thanat", 10000);  
market.addInvestor("Ronnachai", 10000);  
market.addInvestor("Peerawich", 10000);  
market.addInvestor("Apichart", 9000);  
market.addInvestor("Piampong", 9000);  
market.addInvestor("Phitchayarat", 9000);  
market.addInvestor("Goravit", 8000);  
market.addInvestor("Kittinan", 8000);  
market.addInvestor("Waritthon", 8000);  
market.addInvestor("Parinya", 8000);
```

3. สมมุติว่าเรามีหุ้นอยู่ 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 Thanet มีเงิน 10000 บาทแต่แรก ในขั้นนี้ Thanet ก็จะสุ่มเข้าซื้อหุ้นต่าง ๆ ด้วยเงินประมาณครึ่งหนึ่ง การทำแบบนี้อาจารย์เขียนโค้ดไว้ในฟังก์ชันที่ชื่อว่า

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

- ขั้นตอนต่อไปเป็นการตรวจสอบว่า นักลงทุนแต่ละคนมีรายละเอียดบัญชีหุ้น หรือ พอร์ต (Portfolio) ตัวเองเป็นอย่างไร สมมุติว่า Ronnachai และ Thanet อยากทราบว่าตอนนี้พอร์ตตัวเองหน้าตาเป็นอย่างไร ก็จะพิมพ์คำสั่งดังต่อไปนี้

```
market.portfolio("Ronnachai");  
market.portfolio("Thaneat");
```

ผลลัพธ์ก็จะแสดงว่า เหลือเงินในบัญชีที่จะซื้อหุ้นเพิ่มได้อีกกี่บาท และมีหุ้นของบริษัทต่าง อย่างละกี่หุ้น และมีมูลค่าหุ้นรวมเท่าไร

Portfolio of 'Ronnachai', 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 'Thaneat', 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 Ronnchai 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 Thaneat 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 Thaneat 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 ครั้งทีรันใหม่ (สังเกตด้วยการรันคำสั่ง market.showStockOwnership()) เพื่อให้การบ้านนี้ง่ายต่อการคาดเดาผลลัพธ์ อาจารย์ได้สร้างอีกฟังก์ชันหนึ่งซึ่งจะกำหนดจำนวนหุ้นเริ่มแรกคงที่ทุกครั้งที่รันใหม่ ชื่อว่า

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

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

8. ต่อไปเป็นการเสนอขาย [นักศึกษาต้องทำการ implement Heap เพื่อรองรับการทำงานตั้งแต่นี้เป็นต้นไป] จากตารางข้อ 6 นักศึกษาสังเกตว่า จะมีนักลงทุนแค่สองคนเท่านั้น ที่ถือหุ้นของธนาคารไทยพาณิชย์ (SCB) คือ Peerawich 10 หุ้น และ Goravit 10 หุ้น ทั้งสองคนเห็นว่าตัวเองถือหุ้นของบริษัทชั้นดี มีการเติบโตในอนาคตสูง และเป็นที่ต้องการของนักลงทุนคนอื่น จึงจะตั้งราคาขายไว้สูง Peerawich จึงส่งคำสั่งเสนอขายหุ้น SCB ไว้ที่หุ้นละ 200 บาท จำนวน 5 หุ้น (ไม่ยอมขายหมด) ส่วน Goravit เสนอขายที่หุ้นละ 180 บาท จำนวน 10 หุ้น (ขายหมดเลย ถูกกว่าด้วย) ทั้งสอง (ซึ่งปกติจะไม่รู้จักกัน) จึงต่างคนต่างส่งคำสั่งไปที่ระบบ AOM ดังนี้

```
market.submitSellOrder("Peerawich", "SCB", 5, 200);  
market.submitSellOrder("Goravit", "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 Apichart และ Piampong ซึ่งตอนนี้ยังไม่มีหุ้น SCB เห็นศักยภาพของบริษัท SCB ต้องการจะเป็นเจ้าของ SCB (เป็นผู้ถือหุ้น) อยากได้หุ้น SCB มาถือกันคนละ 6 หุ้น จึงจะเสนอซื้อหุ้น SCB สุราราคาได้เต็มที่ไม่เกิน 250 บาทต่อหุ้น ทั้งสองจึงส่งคำสั่งดังนี้

```
market.submitBuyOrder("Apichart", "SCB", 6, 250);  
market.submitBuyOrder("Piampong", "SCB", 6, 250);
```

ผลลัพธ์จะแสดงดังนี้

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

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

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

จะเห็นว่า ราคาเสนอซื้อที่ทั้งสองคนส่งเข้าระบบ ได้ Match กับราคาเสนอขายของทั้ง Goravit และ Peerawich ผลก็คือหุ้น SCB 6 หุ้นของ Goravit จึงแลกเปลี่ยนกับเงินสดของ Apichart โดยมูลค่าเงินที่ Apichart ต้องส่งเข้าบัญชีของ Goravit จะ

เท่ากับ $(\text{จำนวนหุ้น} \times \text{ราคาที่ตั้งรอ}) = (6 \times 180) = 1080$ บาท หลังจากเหตุการณ์นี้ คำสั่งเสนอขายของ Goravit จึงถูกปรับปรุงจากเสนอขาย 10 หุ้นเหลือเสนอขาย 4 หุ้น

นักศึกษาจะสังเกตเห็นได้ว่า ถึงแม้ว่า Goravit จะส่งคำสั่งเสนอขายเข้ามาที่หลัง Peerawich แต่กลับได้สิทธิ์ในการขายออกไปก่อน นั่นก็เพราะว่า ในคิวคำสั่งเสนอขาย ราคาต่ำกว่าจะมี Priority ที่สูงกว่าราคาเสนอขายที่สูงกว่าเสมอครับ คำสั่งขายของ Goravit จึงแซงคิว Peerawich ได้ทันที (ตรงนี้ นศ มองเห็น Min Heap ใช้ไม่ครับ)

ส่วนคำสั่งซื้อ 6 หุ้นของ Piampong ซึ่งส่งคำสั่งเข้ามาช้ากว่า จึง Match กับ 4 หุ้นที่เหลือของ Goravit (ที่ราคา 180 ตามที่ Goravit ตั้งราคารอขายไว้) และอีก 2 หุ้นที่เหลือ จึงเข้าไป Match กับคิวถัดไปนั่นก็คือ Peerawich (ที่ราคา 200 บาทต่อหุ้น) สรุป Piampong ได้รับหุ้น 6 หุ้นตามหวังและต้องส่งเงินจำนวน $4 \times 180 = 720$ เข้าบัญชี Goravit และส่งเงิน $2 \times 200 = 400$ บาทเข้าบัญชี Peerawich

สังเกตว่า คำสั่งเสนอขายของ Peerawich ยังไม่สิ้นสุด ยังเหลือ เสนอขายอีก 3 หุ้นสุดท้าย หากมีคนต้องการซื้อจะสามารถซื้อได้ 3 หุ้นนี้เท่านั้น ที่ราคา 200 บาท ยกเว้นแต่ถ้าจะมีใครส่งคำสั่งเสนอขายเข้ามาอีก หรือถ้าเสนอขายที่ราคาต่ำกว่านี้ เช่น 150 บาทต่อหุ้น คิวการเสนอขายของ Peerawich ก็จะถูกผลักลง Heap โดยทันที

ถ้าหากเราต้องการตรวจบัญชีปัจจุบันของตัวละครทั้งสี่คน เราสามารถใช้คำสั่งนี้

```
market.portfolio("Apichart");
market.portfolio("Piampong");
market.portfolio("Peerawich");
market.portfolio("Goravit");
```

ซึ่งจะให้ผลลัพธ์ดังนี้

Portfolio of 'Apichart', 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 'Piampong', 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 'Peerawich', 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 'Goravit', 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

จะเห็นว่า เงินสดของ Apichart และ Piampong ลดลงเมื่อเทียบกับเงินต้นแต่แรก แต่ทั้งสองคนนี้ก็ยังมีหุ้น SCB เข้ามาอยู่ในพอร์ตกันคนละ 6 หุ้น ส่วนหุ้น SCB ของ Peerawich และ Goravit ก็ขายไปตัวจำนวน (แต่ก็มีเงินสดเพิ่มขึ้นจากการขายหุ้น)

11. ต่อไปเป็นการตั้งราคาเสนอซื้อให้มีหลาย ๆ คนบ้าง จากตารางข้อ 6 นักศึกษาจะสังเกตว่าหุ้นของธนาคารกสิกรไทย (KBANK) ขายตลาดมาก ๆ ทุกคนในตลาดอยากได้หุ้นบริษัทนี้ แต่ไม่ยอมซื้อที่ราคาสูง เพราะจากการประเมินงบการเงินอะไรต่าง ๆ แล้ว หุ้น KBANK ควรจะมีราคาอยู่ประมาณ 100 บาทเท่านั้น Phitchayarat เป็นคนแรกที่เข้ามาเช็คราคาในตลาด ด้วยคำสั่ง `market.quote("KBANK")`;

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

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

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

ผลลัพธ์

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

ต่อมานักลงทุน Goravit, Kittinan, Waritthon, Parinya ต่างก็อยากถือหุ้น KBANK จึงเข้ามาเสนอซื้อที่ราคาต่าง ๆ ดังนี้

```
market.submitBuyOrder("Goravit", "KBANK", 1, 85);  
market.quote("KBANK");  
market.submitBuyOrder("Kittinan", "KBANK", 3, 90);  
market.quote("KBANK");  
market.submitBuyOrder("Waritthon", "KBANK", 2, 100);  
market.quote("KBANK");  
market.submitBuyOrder("Parinya", "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 Apichart เป็นเพียงคนเดียวที่มีหุ้น KBANK อยู่ในมือ, Apichart ตระหนักดีว่า ตัวเองซื้อหุ้นนี้มาตอนที่ราคา 170 บาท บัดนี้ตลาดเสนอราคาซื้อ (ที่ดีที่สุด) แค่ 100 บาท/หุ้น Apichart มีทางเลือกที่จะไม่ขายหุ้นนี้ก็ได้ (ถือกินเงินปันผล รอบริษัทโตไปเรื่อย ๆ) หรือ รอให้คนที่เสนอซื้อให้ราคาสูงกว่านี้ในราคาที่ Apichart รับผิดชอบแล้วค่อยขาย หรือ Apichart อาจจะเสนอขายเองที่ราคาที่สูงกว่านี้ก็ได้ เช่น

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

ผลลัพธ์ก็จะแสดงว่า

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

ซึ่งมันอาจจะเป็นอย่างนี้ตราบนานเท่านานก็ได้ ตราบใดที่ยังไม่มีใครเสนอราคาซื้อที่สูงกว่า 150 บาท

สมมติเราเพิ่มเติมอีกหน่วยก็แล้วกัน ให้ Apichart อยู่ในสถานะร้อนเงิน หุ้น KBANK เป็นหุ้นที่ไม่ดี Apichart จึงตัดสินใจขาย แต่ขายแค่ 6 หุ้น และจะขายในราคาที่ไม่ต่ำกว่า 85 บาทต่อหุ้น Apichart จึงส่งคำสั่งดังต่อไปนี้

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

ผลลัพธ์จึงเป็นดังนี้

```
-----  
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Apichart' to 'Waritthon'  
Also, 200.0 baht is transferred from 'Waritthon' to 'Apichart'  
-----  
-----
```

```
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Apichart' to 'Phitchayarat'  
Also, 180.0 baht is transferred from 'Phitchayarat' to 'Apichart'  
-----  
-----
```

```
Matched!!! Now 2 shares of Stock KBANK are transferred from 'Apichart' to 'Kittinan'  
Also, 180.0 baht is transferred from 'Kittinan' to 'Apichart'  
-----  
-----
```

เนื่องจากมีผู้มารอซื้อจำนวนมาก จึงเกิดคำสั่ง Match ขึ้นมากมาย ในกรณีนี้ Apichart เสนอขายในราคาค่อนข้างต่ำคือ 85 บาทต่อหุ้น จึงเกิดการ Match ทั้งหมด (แต่คนที่มารอซื้อ ไม่ได้ทั้งหมด เพราะหุ้นที่เสนอขายมีน้อย) Waritthon ได้เสนอซื้อในราคาที่สูงที่สุด คือ 100 บาทต่อหุ้น จึงได้สิทธิ์แลกเปลี่ยน 2 หุ้นที่เสนอซื้อเข้ามาก่อน ได้ไปก่อน จึงโอนเงินไปให้ $2 \times 100 = 200$ บาท, คนต่อมาคือ Phitchayarat ซึ่งเสนอราคาซื้อไว้ที่ 90 บาทต่อหุ้น (และส่งคำสั่งเสนอมาก่อนคู่แข่งคือ Kittinan) จึงขายออกได้ 2 หุ้น โดยแลกกับเงิน $2 \times 90 = 180$ บาท ส่วน Kittinan เสนอซื้อราคาที่ 90 บาทเหมือนกัน แต่ช้ากว่า จึงได้ที่หลัง ถึงแม้ว่า Kittinan จะเสนอซื้อ 3 หุ้นแต่ก็ได้ไปแค่ 2 หุ้นเพราะ Apichart เสนอขายแค่ 6 หุ้น (ให้ไปกับคนก่อนหน้าแล้ว 4 จึงเหลือขายให้ Kittinan แค่ 2 หุ้น) เพราะฉะนั้นคำสั่งที่ค้างเหลืออยู่ในระบบตอนนี้จึงเป็น คำสั่งซื้อของ Kittinan 1 หุ้นที่ราคา 90 บาท, คำสั่งซื้อของ Parinya 2 หุ้นที่ราคา 80 บาท และสุดท้ายคำสั่งขายของ Apichart เอง 2 หุ้นแต่แรก ที่เสนอขายไว้ 150 บาท ถึงปานนี้ก็ยังไม่มีใครจะซื้อที่ราคานี้ หากตรวจสอบ Portfolio ของ Apichart จะพบว่ามียาละเอียดดังนี้

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

```
-----  
Portfolio of 'Apichart', 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 Apichart มีหุ้น KBANK ลดลงจาก 10 หุ้นเหลือ 4 หุ้น (เพราะขายไปเมื่อ 6 หุ้น) สังเกตว่าสินทรัพย์โดยรวม (Total portfolio values) มีมูลค่าลดลง เนื่องจากราคาหุ้นตก จากที่ซื้อมา 170 บาทต่อหุ้น ตอนนี้เหลือราคาซื้อขายกันที่ 85 บาทต่อหุ้น เงินจึงหายวับจากที่ราคาหุ้นผันผวน อย่างไรก็ตาม Apichart ก็บอกกับเพื่อน ๆ ว่า 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("Thaneat", "PTT", 3, 350);
market.submitSellOrder("Thaneat", "PTT", 3, 360);
market.submitSellOrder("Thaneat", "PTT", 4, 370);
market.submitSellOrder("Peerawich", "PTT", 5, 355);
market.submitSellOrder("Peerawich", "PTT", 5, 365);
market.submitSellOrder("Peerawich", "PTT", 5, 375);
market.submitSellOrder("Piampong", "PTT", 3, 360);
market.submitSellOrder("Piampong", "PTT", 3, 370);
market.submitSellOrder("Piampong", "PTT", 4, 380);

market.submitBuyOrder("Ronnachai", "PTT", 6, 370);
market.submitBuyOrder("Apichart", "PTT", 6, 370);
market.submitBuyOrder("Phitchayarat", "PTT", 10, 370);
market.submitBuyOrder("Goravit", "PTT", 10, 370);

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

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

market.submitSellOrder("Parinya", "PTT", 3, 320);
market.submitSellOrder("Parinya", "PTT", 3, 240);
market.submitSellOrder("Parinya", "PTT", 4, 220);
market.submitSellOrder("Ronnachai", "PTT", 3, 320);
market.submitSellOrder("Ronnachai", "PTT", 3, 240);
market.submitSellOrder("Ronnachai", "PTT", 4, 220);

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

Matched!!! Now 3 shares of Stock PTT are transferred from 'Thaneat' to 'Ronnachai'
Also, 1050.0 baht is transferred from 'Ronnachai' to 'Thaneat'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Peerawich' to 'Ronnachai'
Also, 1065.0 baht is transferred from 'Ronnachai' to 'Peerawich'

Matched!!! Now 2 shares of Stock PTT are transferred from 'Peerawich' to 'Apichart'
Also, 710.0 baht is transferred from 'Apichart' to 'Peerawich'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Thaneat' to 'Apichart'
Also, 1080.0 baht is transferred from 'Apichart' to 'Thaneat'

Matched!!! Now 1 shares of Stock PTT are transferred from 'Piampong' to 'Apichart'
Also, 360.0 baht is transferred from 'Apichart' to 'Piampong'

Matched!!! Now 2 shares of Stock PTT are transferred from 'Piampong' to 'Phitchayarat'
Also, 720.0 baht is transferred from 'Phitchayarat' to 'Piampong'

Matched!!! Now 5 shares of Stock PTT are transferred from 'Peerawich' to 'Phitchayarat'
Also, 1825.0 baht is transferred from 'Phitchayarat' to 'Peerawich'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Thaneat' to 'Phitchayarat'
Also, 1110.0 baht is transferred from 'Phitchayarat' to 'Thaneat'

Matched!!! Now 1 shares of Stock PTT are transferred from 'Thaneat' to 'Goravit'

Also, 370.0 baht is transferred from 'Goravit' to 'Thaneat'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Piampong' to 'Goravit'

Also, 1110.0 baht is transferred from 'Goravit' to 'Piampong'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Parinya' to 'Goravit'

Also, 1110.0 baht is transferred from 'Goravit' to 'Parinya'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Parinya' to 'Goravit'

Also, 1110.0 baht is transferred from 'Goravit' to 'Parinya'

Matched!!! Now 2 shares of Stock PTT are transferred from 'Parinya' to 'Kittinan'

Also, 660.0 baht is transferred from 'Kittinan' to 'Parinya'

Matched!!! Now 2 shares of Stock PTT are transferred from 'Parinya' to 'Waritthon'

Also, 660.0 baht is transferred from 'Waritthon' to 'Parinya'

Matched!!! Now 1 shares of Stock PTT are transferred from 'Ronnachai' to 'Waritthon'

Also, 330.0 baht is transferred from 'Waritthon' to 'Ronnachai'

Matched!!! Now 2 shares of Stock PTT are transferred from 'Ronnachai' to 'Kittinan'

Also, 600.0 baht is transferred from 'Kittinan' to 'Ronnachai'

Matched!!! Now 1 shares of Stock PTT are transferred from 'Ronnachai' to 'Kittinan'

Also, 270.0 baht is transferred from 'Kittinan' to 'Ronnachai'

Matched!!! Now 1 shares of Stock PTT are transferred from 'Ronnachai' to 'Kittinan'

Also, 270.0 baht is transferred from 'Kittinan' to 'Ronnachai'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Ronnachai' to 'Waritthon'

Also, 750.0 baht is transferred from 'Waritthon' to 'Ronnachai'

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("Ronnachai");		

Portfolio of 'Ronnachai', 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 Ronnachai ได้หายไป 10000(จากเดิมตั้งต้นเลย) – 7185(ปัจจุบัน) = 2815 บาท
นักศึกษาสามารถตอบได้หรือไม่ว่าทำไม หายไปไหน?

14. สมมุติว่าจากเหตุการณ์ข้อ 13 มีการซื้อขายกับอีกเล็กน้อยดังนี้

market.submitBuyOrder("Thaneat", "PTT", 20, 150);	
market.submitBuyOrder("Thaneat", "PTT", 20, 100);	
market.submitBuyOrder("Thaneat", "PTT", 10, 50);	
market.submitSellOrder("Phitchayarat", "PTT", 10, 125);	
market.submitSellOrder("Goravit", "PTT", 10, 75);	
market.submitSellOrder("Kittinan", "PTT", 15, 75);	
market.submitSellOrder("Waritthon", "PTT", 15, 75);	
market.quote("PTT");	

ให้ผลลัพธ์คือ

Matched!!! Now 3 shares of Stock PTT are transferred from 'Phitchayarat' to 'Waritthon'
Also, 600.0 baht is transferred from 'Waritthon' to 'Phitchayarat'

Matched!!! Now 7 shares of Stock PTT are transferred from 'Phitchayarat' to 'Thaneat'
Also, 1050.0 baht is transferred from 'Thaneat' to 'Phitchayarat'

Matched!!! Now 10 shares of Stock PTT are transferred from 'Goravit' to 'Thaneat'
Also, 1500.0 baht is transferred from 'Thaneat' to 'Goravit'

Matched!!! Now 3 shares of Stock PTT are transferred from 'Kittinan' to 'Thaneat'
Also, 450.0 baht is transferred from 'Thaneat' to 'Kittinan'

Matched!!! Now 12 shares of Stock PTT are transferred from 'Kittinan' to 'Thaneat'
Also, 1200.0 baht is transferred from 'Thaneat' to 'Kittinan'

Matched!!! Now 8 shares of Stock PTT are transferred from 'Waritthon' to 'Thaneat'
Also, 800.0 baht is transferred from 'Thaneat' to 'Waritthon'

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

คุณสังเกตได้หรือไม่ว่า Ronnachai ไม่ได้เกี่ยวข้องกับการซื้อขายกันในข้อ 14. เลข คราวนี้มาดู Portfolio ของ Ronnachai กัน

market.portfolio("Ronnachai");		

Portfolio of 'Ronnachai', 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 Ronnachai ลดลงอีก เหลือ 4575 เงินหายไป $10000 - 4575 = 5425$ หรือ -54% นักศึกษาทราบหรือไม่ว่า เงิน Ronnachai หายไปไหน ทั้ง ๆ ที่ไม่ได้เกี่ยวข้องกับการซื้อขายหุ้นเลย ผมจะไปถามอีกทีตอนสอบครับ

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

ภาคผนวก

Test Case ดังต่อไปนี้ ไม่ได้ตรวจคะแนน แต่จะช่วยให้ นักศึกษา เข้าใจการทำงานของ Priority Queue ได้ดีขึ้น อาจารย์อยากให้นักศึกษาลองแก้โค้ดฟังก์ชัน `public boolean compare(Node rightNode)` ซึ่งเป็นสมาชิกของ class `Node` ฟังก์ชันนี้จะทำการเปรียบเทียบ Priority ของสอง Node ใด ๆ ว่า Node ทางด้านซ้ายมี Priority สูงกว่า Node ทางด้านขวาหรือไม่ เช่น Node A เป็นสมาชิกของ Min Heap มีราคา 30 บาท เข้าคิวมาตอน Timestamp เป็น 1 ส่วน Node B มีราคา 20 บาท เข้าคิวมาตอน Timestamp เป็น 2 ด้วยเงื่อนไขนี้ Node B จะมี Priority สูงกว่า Node A เพราะ Min Heap กำหนดให้ราคาน้อยกว่ามี Priority ที่สูงกว่าเสมอ แต่ถ้าทั้งสอง Node มีราคาเท่ากัน ให้ดูที่เวลา ใครมาก่อนจะมี Priority ที่สูงกว่า ในการบ้านนี้เวลาจะไม่ซ้ำกัน ทำให้ สอง Node ใด ๆ จะมี ใครสักคนที่มี Priority สูงกว่าเสมอ

ตัวอย่างการใช้งาน ฟังก์ชัน `compare` คือ `nodeA.compare(nodeB)` โดยที่ `nodeA` และ `nodeB` เป็นตัวแปรชี้ไปยัง Object ของ Node A และ Node B ตามลำดับ ผลลัพธ์จะเป็นจริง ถ้า Node A มี Priority สูงกว่า Node B และเมื่อนักศึกษาได้แก้ไขฟังก์ชัน `compare` ของอาจารย์ได้เสร็จสมบูรณ์แล้ว Test Case ดังต่อไปนี้ควรที่จะทำงานได้ถูกต้อง

Java code
<pre>public static void main(String[] args) { Node nodeA = new Node(); Node nodeB = new Node(); nodeA.price = 30; nodeA.timestamp = 1; nodeA.isMinHeap = true; nodeB.price = 20; nodeB.timestamp = 2; nodeB.isMinHeap = true; System.out.println("nodeA.compare(nodeB) = " + nodeA.compare(nodeB)); System.out.println("nodeB.compare(nodeA) = " + nodeB.compare(nodeA)); nodeA.price = 30; nodeA.timestamp = 1; nodeA.isMinHeap = false; // Max Heap nodeB.price = 20; nodeB.timestamp = 2; nodeB.isMinHeap = false; // Max Heap System.out.println("nodeA.compare(nodeB) = " + nodeA.compare(nodeB)); System.out.println("nodeB.compare(nodeA) = " + nodeB.compare(nodeA)); nodeA.price = 100; nodeA.timestamp = 1; nodeA.isMinHeap = true; nodeB.price = 100; nodeB.timestamp = 2; nodeB.isMinHeap = true; System.out.println("nodeA.compare(nodeB) = " + nodeA.compare(nodeB)); System.out.println("nodeB.compare(nodeA) = " + nodeB.compare(nodeA)); }</pre>
Output
<pre>nodeA.compare(nodeB) = false nodeB.compare(nodeA) = true nodeA.compare(nodeB) = true nodeB.compare(nodeA) = false nodeA.compare(nodeB) = true nodeB.compare(nodeA) = false</pre>