Homework: Stock Market Simulation using Priority Queues

Patiwet Wuttisarnwattana, Ph.D.
Department of Computer Engineering
Chiang Mai University

การบ้านนี้นักศึกษาจะได้เรียนรู้การทำงานของตลาดหลักทรัพย์ ซึ่งเป็นแหล่งแลกเปลี่ยน/ซื้อขายหุ้นกันระหว่างนักลงทุน เช่น สมมุติ นักลงทุน A มีหุ้นของบริษัท XYZ อยู่จำนวนหนึ่ง ต้องการขายหุ้นที่มีออกบ้าง (หรือทั้งหมด) เพราะต้องการใช้เงินด่วน จึงตัดสินใจส่งคำสั่งเสนอขาย (Offer) ไปยังระบบซื้อขายของตลาดหลักทรัพย์ นักลงทุน A จะขายหุ้นออกไม่ได้เลยจนกว่าจะมีคนที่สนใจหุ้นที่ว่า ต่อมาไม่นาน นักลงทุน B ซึ่งต้องการเข้าถือหุ้น (มีส่วนในการเป็นเจ้าของ) บริษัท 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 มีคำสั่งซื้อขายเข้ามานับแสนนับล้านคำสั่ง แต่ละคนก็จะเสนอซื้อและเสนอขายในราคาที่แตกต่างกัน ดังนั้นระบบจะต้องเรียงคำสั่งเสียใหม่เพื่อหาว่าคำสั่งใดจะได้ไปก่อนไปหลังตามลำดับที่ควรจะเป็น

หลักใหญ่ใจความของระบบการซื้อขายในตลาดหลักทรัพย์ก็จะคล้ายกับระบบประมูล คือใครเสนอราคาที่ดีที่สุดเข้ามาก่อน คนนั้นก็จะได้รับสิทธิ์ในการซื้อขายไป เพียงแต่ระบบประมูลที่ว่าจะมีทั้งคนเสนอขายและเสนอซื้อ คนเสนอขายที่ขายราคาต่ำที่สุดจะได้สิทธิ์ก่อน คนเสนอซื้อที่ซื้อราคาสูงที่สุดจะได้สิทธิก่อน ส่วนคนที่ส่งคำสั่งที่ราคาเดียวกัน คนที่ส่งมาก่อนจะได้สิทธิ์ก่อน นักศึกษาจะเห็นได้ว่านี่มัน Priority Queue ชัด ๆ

ในลำดับถัดไป อาจารย์จะขออธิบายวิธีการที่ระบบจะจัดเรียงคำสั่งที่รับเข้ามาหลาย ๆ คำสั่ง ดังนี้

1. การจัดเรียงลำดับคำสั่งซื้อขายเมื่อสามารถส่งคำสั่งซื้อขายเข้ามา ระบบการซื้อขายจะเก็บคำสั่งซื้อขายไว้ตั้งแต่เวลาที่ส่งคำสั่งซื้อขาย จนถึงสิ้นวันทำการ และจัดเรียงคำสั่งซื้อขายตามลำดับของราคาและเวลาที่ดีที่สุด (Price then Time Priority) โดยมีหลักการคือ
   1.1. คำสั่งซื้อที่มีราคาเสนอซื้อ สูงที่สุด จะถูกจัดเรียงไว้ในลำดับที่หนึ่ง และถ้ามีราคาเสนอซื้อที่สูงกว่าถูกส่งเข้ามาใหม่ จะจัดเรียงราคาเสนอซื้อที่สูงกว่าเป็นการเสนอซื้อในลำดับแรกก่อน และ ถ้ามีการเสนอซื้อในแต่ละราคามากกว่าหนึ่งรายการ (เสนอที่ราคาเท่ากัน แต่ส่งคำสั่งเข้ามาหลายรายการ) ให้จัดเรียงตามเวลา โดยคำสั่งที่ส่งเข้ามาก่อน จะถูกจัดไว้เป็นการเสนอซื้อในลำดับต้น ๆ
   1.2. คำสั่งขายที่มีราคาเสนอขาย ต่ำที่สุด จะถูกจัดเรียงไว้ในลำดับที่หนึ่ง และถ้ามีราคาเสนอขายที่ต่ำกว่าถูกส่งเข้ามาใหม่ จะจัดเรียงราคาเสนอขายที่ต่ำกว่าเป็นการเสนอขายในลำดับแรกก่อน และ ถ้ามีการเสนอขายในแต่ละ

ราคามากกว่าหนึ่งรายการ (เสนอที่ราคาเท่ากัน แต่ส่งคำสั่งเข้ามาหลายรายการ) ให้จัดเรียงตามเวลา โดยคำสั่งที่ส่งเข้ามาก่อน จะถูกจัดไว้เป็นการเสนอขายในลำดับต้น ๆ

2. การจับคู่การซื้อขาย (Matching) เมื่อมีคำสั่งซื้อขายส่งเข้ามาในระบบแล้ว ระบบจะตรวจสอบว่าคำสั่งนั้นสามารถจับคู่กับคำสั่งด้านตรงข้ามได้ทันทีหรือไม่ ถ้าคำสั่งนั้นสามารถจับคู่ได้ทันที ระบบก็จะทำการจับคู่ให้ แต่ถ้าคำสั่งนั้น ไม่สามารถจับคู่ได้ ระบบจะส่งคำสั่งนั้นไปรอในระบบ Priority Queue โดยลำดับความสำคัญจะเป็นไปตามหลักการ Price then Time Priority เงื่อนไข ในการจับคู่กันได้ของคำสั่งก็คือ หากราคาเสนอซื้อที่ดีที่สุด (สูงที่สุด) มีค่ามากกว่าหรือเท่ากับ ราคาเสนอขายที่ดีที่สุด (ต่ำที่สุด) การจับคู่กันของคำสั่งก็จะเกิดขึ้น

ในการบ้านนี้อาจารย์ได้ทำการ implement ระบบตลาดหลักทรัพย์ไว้หมดแล้ว ทั้งการเพิ่มนักลงทุน, การเพิ่มหุ้น, การกระจายหุ้นให้นักลงทุนตอนเริ่มแรก, การส่งคำสั่งซื้อ/คำสั่งขาย, การจับคู่กันได้ของคำสั่ง, การโอนหุ้นและโอนเงินเมื่อเกิดการจับคู่, การแสดงข้อมูล Portfolio, การแสดงข้อมูล Quote, และอื่น ๆ **โดยหน้าที่ของนักศึกษา คือทำการศึกษาและทำความเข้าใจในสิ่งที่อาจารย์เขียน** แต่ช้าก่อนอย่าเพิ่งไปอ่านโค้ดอาจารย์ครับ โดยขอคุณอ่านการบรรยายโจทย์และการจำลองสถานการณ์ของอาจารย์ตามเอกสารฉบับนี้ให้เข้าใจเสียก่อน แล้วจึงค่อยไปเริ่มอ่านโค้ดอาจารย์ ถ้าสิ่งเหล่านี้เป็นเรื่องที่เกินความสามารถของนักศึกษา ก็ให้ประชุมกัน ให้เพื่อนช่วยติวให้ ฝึกวิเคราะห์ตีความว่าโค้ดตรงนั้นตรงนี้อาจารย์น่าจะหมายความว่าอย่างไร ในอนาคตนักศึกษาก็จะต้องไปศึกษาโค้ดของคนอื่น ๆ ครับ ดังนั้น การบ้านนี้จะฝึกให้ นศ มีความสามารถในการอ่านโค้ดของผู้อื่นให้เป็น

อย่างไรก็ตาม ระบบทั้งหมดในการบ้านนี้ เกือบจะเสร็จหมดแล้ว ยกเว้นเหลือเพียงระบบเดียวคือระบบ Priority Queues ที่จะทำการเรียงลำดับความสำคัญของคำสั่งที่ระบบส่งเข้ามาให้กับคลาส Heap โดย**นักศึกษาจะต้อง implement ระบบดังกล่าวด้วย Binary Heap โดยใช้วิธีการของ Array-as-a-Complete-Binary-Tree** แค่นั้นเอง

ในการบ้านนี้เราจะมี class StockMarket จะทำหน้าที่ดำเนินการเป็นระบบตลาดหลักทรัพย์ ที่จะบันทึกข้อมูลของนักลงทุน (class Investor), ข้อมูลของหุ้น (class Stock), การประสานงานกับระบบ Priority Queues ผ่าน class Heap, ระบบบัญชีที่บันทึกว่านักลงทุนแต่ละคนมีหุ้นแต่ละบริษัทกี่หุ้น (ผ่านตัวแปรที่ชื่อว่า int[][] stockOwnership) อย่างไรก็ตาม ระบบที่อาจารย์เขียนขึ้นนี้เป็นเหตุการณ์สมมุติ ระบบในตลาดหลักทรัพย์ที่ใช้จริงมีความสลับซับซ้อนกว่านี้หลายเท่าตัว ที่คุณจะได้ศึกษาต่อไปในการบ้านนี้เป็นแค่แนวคิดอย่างง่ายเท่านั้น

อาจารย์ขอเริ่มต้นอธิบายการบ้านต่อไป ด้วยเหตุการณ์จำลองดังต่อไปนี้

1. สมมุติว่าเราต้องการสร้าง Object ของ StockMarket โดยรองรับนักลงทุนได้ 10 คน มีหุ้นบริษัทอยู่ 10 บริษัท โค้ดของเราก็จะเขียนได้ว่า

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

2. สมมุติว่าเรามีนักลงทุน 10 คนดังต่อไปนี้

รหัสนักลงทุน (Auto-generated)	ชื่อนักลงทุน	เงินทุนเริ่มแรก (บาท)
0	Matthew	10000
1	Mark	10000
2	Luke	10000

3	John	9000
4	James	9000
5	Peter	9000
6	Jude	8000
7	Paul	8000
8	Timothy	8000
9	Titus	8000

คำสั่งการเพิ่มนักลงทุนก็จะเขียนได้ว่า

```
market.addInvestor("Matthew", 10000); // ID=0
market.addInvestor("Mark", 10000); // ID=1
market.addInvestor("Luke", 10000); // ID=2
market.addInvestor("John", 9000); // ID=3
market.addInvestor("James", 9000); // ID=4
market.addInvestor("Peter", 9000); // ID=5
market.addInvestor("Jude", 8000); // ID=6
market.addInvestor("Paul", 8000); // ID=7
market.addInvestor("Timothy", 8000); // ID=8
market.addInvestor("Titus", 8000); // ID=9
```

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);
```

4. ขั้นตอนต่อไปเป็นการที่นักลงทุนจะเข้าซื้อหุ้นกับบริษัทโดยตรงในตลาดแรก (หรือที่รู้จักกันในวงการคือ Initial Public Offering (IPO)) โดยอาจารย์จะจำลองสถานการณ์ ให้นักลงทุนแต่ละคนซื้อหุ้นเข้าพอร์ตตัวเองโดยใช้เงินประมาณ 5,000 บาท (จากเงินตั้งต้นคนละ 10,000 บาท) โดยการสุ่ม (เพื่อให้แต่ละคนมีหุ้นในพอร์ตลงทุนไม่เหมือนกัน) อาจารย์ได้เขียนโค้ดเพื่อจำลองสถานการณ์นี้เอาไว้ในฟังก์ชันที่ชื่อว่า

```
market.giveMeSomeShares ();
```

5. ขั้นต่อไปเป็นจะการตรวจสอบว่า นักลงทุนต่อละคนมีรายละเอียดบัญชีหุ้น หรือ พอร์ตการลงทุน (Portfolio) ตัวเองเป็นอย่างไร สมมุติว่า Mark และ Matthew อยากรู้ว่าตอนนี้พอร์ตการลงทุนตัวเองหน้าตาเป็นอย่างไร ก็จะพิมพ์คำสั่งดังต่อไปนี้

```
market.portfolio("Mark");
market.portfolio("Matthew");
```

ผลลัพธ์ก็จะแสดงว่า นักลงทุนรายนี้ มีหุ้นในบริษัทใดบ้าง จำนวนกี่หุ้น มีเงินสดเหลือ (ที่จะซื้อหุ้นเพิ่มได้อีก) กี่บาท และทั้งบัญชีนี้ มีมูลค่าสุทธิรวม (มูลค่าหุ้นรวมทุกบริษัท + เงินสด) เท่าไหร่

```
-------------------------------------------
  Portfolio of 'Mark', 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
```

```
-------------------------------------------
  Portfolio of 'Matthew', 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	0	0.0	TRUE	20	143.0
CPN	0	0.0	CPN	10	530.0
BTS	0	0.0	BTS	0	0.0
DTAC	0	0.0	DTAC	10	355.0
LH	20	175.0	LH	0	0.0
---			---		
Total portfolio values = 10000.0 bath			Total portfolio values = 10000.0 bath		
---			---		

จากตัวอย่าง Mark 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 Matthew อาจารย์เข้าใจว่านักศึกษาสามารถอ่านเองได้ว่าหมายถึงอะไร

6. ขั้นต่อไปเป็นการตรวจสอบทั้งตลาดเลยว่า 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 มีหุ้นบริษัทต่าง ๆ ซึ่งอยู่แนวตั้ง อย่างละกี่หุ้น ยกตัวอย่างเช่น นักลงทุน ID [0] ซึ่งก็คือ Matthew มีหุ้น PTT 10 หุ้น, มีหุ้น CPALL อยู่ 10 หุ้น, มีหุ้น CPF 10 หุ้น, มีหุ้น TRUE 20 หุ้น, มีหุ้น CPN และหุ้น DTAC อย่างละ10 หุ้น ส่วนคนอื่น ๆ ก็ใช้หลักการอ่านเดียวกัน

7. นักศึกษาจะสังเกตเห็นว่า คำสั่ง market.giveMeSomeShares() จะสุ่มจำนวนหุ้นเข้าพอร์ตทุก ๆ ครั้งที่รันใหม่ (สังเกตด้วยการรันคำสั่ง market.giveMeSomeShares() หลายครั้ง ๆ) เพื่อให้การบ้านนี้ง่ายต่อการคาดเดาผลลัพธ์และการตรวจ อาจารย์ได้สร้างอีกฟังก์ชันหนึ่งซึ่งจะโหลด เซฟที่อาจารย์เคยรันได้ เพื่อให้นักศึกษาทุกคนจะให้ผลลัพธ์ที่ตรงกัน คำสั่งที่ว่านี้คือ

```
market.loadAssets();
```

ผลลัพธ์จากการรันฟังก์ชันนี้ จะให้ผลลัพธ์เดียวกันกับข้อที่ 6. ดังนั้น ขอให้นักศึกษาใช้ ฟังก์ชัน market.loadAssets() นี้แทน ฟังก์ชัน market.giveMeSomeShares() เพื่อตรวจสอบความถูกต้องของตัวเลขให้ตรงกันครับ

8. ต่อไปเป็นการเสนอขาย **[นักศึกษาต้องทำการปรับปรุง class Heap ให้ทำงานได้ถูกต้องก่อน เพื่อให้รองรับการทำงานตั้งแต่จุดนี้เป็นต้นไป]** จากตารางข้อ 6 นักศึกษาสังเกตว่า จะมีนักลงทุนแค่สองคนเท่านั้น ที่ถือหุ้นของธนาคารไทยพาณิชย์ (SCB) คือ Luke 10 หุ้น และ Jude 10 หุ้น ทั้งสองคนเห็นว่าตัวเองถือหุ้นของบริษัทชั้นดี มีการเติบโตในอนาคตสูง และเป็นที่ต้องการของนักลงทุนคนอื่น จึงจะตั้งราคาขายไว้สูง Luke จึงส่งคำสั่งเสนอขายหุ้น SCB ไว้ที่หุ้นละ 200 บาท จำนวน 5 หุ้น (ไม่อยากขายหมด) ส่วน Jude เสนอขายที่หุ้นละ 180 บาท จำนวน 10 หุ้น (ขายหมดเลย ถูกกว่าด้วย) ทั้งสองนักลงทุนซึ่งปกติจะไม่รู้จักกัน จึงต่างคนต่างส่งคำสั่งไปที่ระบบ ดังนี้

```
market.submitSellOrder("Luke", "SCB", 5, 200);
market.submitSellOrder("Jude", "SCB", 10, 180);
```

9. นักลงทุนคนอื่น ๆ ที่สนใจในหุ้น SCB หากต้องการเช็คราคาเสนอขายล่าสุดของ SCB เค้าเหล่านั้นก็สามารถที่จะตรวจสอบ “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) = 0x0.0 ปกติระบบตลาดหลักทรัพย์จะไม่แสดงชื่อให้เห็นว่า ตอนนี้ใครเป็นผู้เสนอราคาที่ดีที่สุดอยู่ แต่จะแสดงแค่ว่าราคาที่ดีที่สุดตอนนี้ คือราคาอะไร ที่จำนวนหุ้นเท่าไหร่ครับ

ทั้งนี้หากมีคนเสนอราคาที่ดีที่สุดเท่ากันหลาย ๆ คน ระบบจะให้คนที่มาก่อนเท่านั้นที่จะเป็นคนอยู่ด้านบนสุดของ Heap ตามหลักการของ Priority Queue

10. ต่อมานักลงทุน John และ James ซึ่งตอนนี้ยังไม่มีหุ้น SCB เห็นศักยภาพของบริษัท SCB ต้องการจะเป็นเจ้าของ SCB (เป็นผู้ถือหุ้น) อยากได้หุ้น SCB มาถือกันคนละ 6 หุ้น จึงจะเสนอซื้อหุ้น SCB สู้ราคาได้เต็มที่ไม่เกิน 250 บาทต่อหุ้น ทั้งสองจึงส่งคำสั่งดังนี้

```
market.submitBuyOrder("John", "SCB", 6, 250);
market.submitBuyOrder("James", "SCB", 6, 250);
```

ผลลัพธ์จะแสดงดังนี้

```
-----------------------------------------------------------------------------------------
Matched!!! Now 6 shares of Stock SCB are transferred from 'Jude' to 'John'
Also, 1080.0 baht is transferred from 'John' to 'Jude'
-----------------------------------------------------------------------------------------
Matched!!! Now 4 shares of Stock SCB are transferred from 'Jude' to 'James'
Also, 720.0 baht is transferred from 'James' to 'Jude'
```

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

จะเห็นได้ว่า ราคาเสนอซื้อที่ทั้งสองคนส่งเข้าระบบ ได้ Match กับราคาที่เสนอขายของทั้ง Jude และ Luke ผลก็คือ หุ้น SCB 6 หุ้นของ Jude จึงแลกเปลี่ยนกับเงินสดของ John โดยมูลค่าเงินที่ John ต้องส่งเข้าบัญชีของ Jude จะเท่ากับ {จำนวนหุ้น x ราคาที่ตั้งรอ} = {6 x 180} = 1080 บาท หลังจากเหตุการณ์นี้ คำสั่งเสนอขายของ Jude จึงถูกปรับปรุงจากจำนวนหุ้นเสนอขาย 10 หุ้นเหลือจำนวนหุ้นเสนอขาย 4 หุ้น

นักศึกษาจะสังเกตเห็นได้ว่า ถึงแม้ว่า Jude จะส่งคำสั่งเสนอขายเข้ามาทีหลัง Luke แต่กลับได้สิทธิ์ในการขายออกไปก่อน นั่นก็เพราะว่า ในคิวคำสั่งเสนอขาย ราคาที่เสนอขายมีค่าถูกกว่า จึงมี Priority ที่สูงกว่า ราคาเสนอขายที่มีค่าแพงกว่าเสมอครับ คำสั่งขายของ Jude จึงแซงคิว Luke ได้ทันที (ตรงนี้ นศ มองเห็น Min Heap ใช่ไม๊ครับ)

ส่วนคำสั่งซื้อ 6 หุ้นของ James ซึ่งส่งคำสั่งเข้ามาช้ากว่า จึง Match กับ 4 หุ้นที่เหลือของ Jude (ที่ราคา 180 ตามที่ Jude ตั้งราคารอขายไว้) และอีก 2 หุ้นที่เหลือ จึงเข้าไป Match กับคิวถัดไปนั่นก็คือ Luke (ที่ราคา 200 บาทต่อหุ้น) สรุป James ได้รับหุ้น 6 หุ้นตามหวังและต้องส่งเงินจำนวน 4 x 180 = 720 เข้าบัญชี Jude และส่งเงิน 2 x 200 = 400 บาทเข้าบัญชี Luke

สังเกตว่า คำสั่งเสนอขายของ Luke ยังไม่สิ้นสุด ยังเหลือ เสนอขายอีก 3 หุ้นสุดท้าย หากมีคนต้องการซื้อ จะสามารถซื้อได้ 3 หุ้นนี้เท่านั้น ที่ราคา 200 บาท ยกเว้นแต่ว่าจะมีใครส่งคำสั่งเสนอขายเข้ามาอีก หรือถ้าเสนอขายที่ราคาต่ำกว่านี้ เช่น 150 บาทต่อหุ้น คิวการเสนอขายของ Luke ก็จะถูกผลักลึกลงไปใน Heap โดยทันที

ถ้าหากเราต้องการตรวจบัญชีปัจจุบันของตัวละครทั้งสี่คน เราสามารถใช้คำสั่งนี้

```
market.portfolio("John");
market.portfolio("James");
market.portfolio("Luke");
market.portfolio("Jude");
```

ซึ่งจะให้ผลลัพธ์ดังนี้

```
---------------------------------------------
    Portfolio of 'John', 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
```

```
---------------------------------------------
    Portfolio of 'James', 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            10              355.0
LH              0               0.0
--------------------------------------------
Total portfolio values = 9120.0 bath
--------------------------------------------

--------------------------------------------
    Portfolio of 'Luke', 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
--------------------------------------------
```

```
DTAC            0               0.0
LH              10              87.5
--------------------------------------------
Total portfolio values = 9080.0 bath
--------------------------------------------

--------------------------------------------
    Portfolio of 'Jude', 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 John และ James ลดลงเมื่อเทียบกับเงินต้นแต่แรก แต่ทั้งสองคนนี้ก็มีหุ้น SCB เข้ามาอยู่ในพอร์ตกันคนละ 6 หุ้น ส่วนหุ้น SCB ของ Luke และ Jude ก็จะหายไปด้วยปริมาณที่เท่ากัน (แต่ก็มีเงินสดเพิ่มขึ้นจากการขายหุ้น)

11. ต่อไปเป็นการตั้งราคาเสนอซื้อให้มีหลาย ๆ คนบ้าง จากตารางข้อ 6 นักศึกษาจะสังเกตว่าหุ้นของธนาคารกสิกรไทย (KBANK) ขาดตลาดมาก ๆ ทุกคนในตลาดอยากได้หุ้นบริษัทนี้ แต่ไม่อยากซื้อที่ราคาสูง เพราะจากการประเมินงบการเงินอะไรต่าง ๆ แล้ว หุ้น KBANK ควรจะมีราคาอยู่ประมาณ 100 บาทเท่านั้น Peter เป็นคนแรกที่เข้ามาเช็คราคาในตลาด ด้วยคำสั่ง market.quote("KBANK");

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

ข้อมูลแสดงว่า ยังไม่มีใครเสนอราคาซื้อและราคาขายแต่อย่างใด (สังเกต 0x ของทั้งสองฝั่ง) Peter จึงเป็นผู้กำหนดราคาคนแรกของตลาดเลย จึงสั่งคำสั่งเสนอซื้อหุ้น KBANK จำนวน 2 หุ้น ที่ราคา 90 บาท (เผื่อว่าจะซื้อได้ถูกกว่าราคาประเมิน)

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

ผลลัพธ์

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

ต่อมานักลงทุน Jude, Paul, Timothy, Titus ต่างก็อยากถือหุ้น KBANK จึงเข้ามาเสนอซื้อที่ราคาต่าง ๆ ดังนี้

```
market.submitBuyOrder("Jude", "KBANK", 1, 85);
market.quote("KBANK");
market.submitBuyOrder("Paul", "KBANK", 3, 90);
market.quote("KBANK");
market.submitBuyOrder("Timothy", "KBANK", 2, 100);
market.quote("KBANK");
market.submitBuyOrder("Titus", "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 John เป็นเพียงคนเดียวที่มีหุ้น KBANK อยู่ในมือ, John ตระหนักดีว่า ตัวเองซื้อหุ้นนี้มาตอนที่ราคา 170 บาท บัดนี้ตลาดเสนอราคาซื้อ (ที่ดีที่สุด) แค่ 100 บาท/หุ้น John มีทางเลือกที่จะไม่ขายหุ้นนี้ก็ได้ (ถือกินเงินปันผล รอบริษัทโตไปเรื่อย ๆ) หรือ รอให้คนที่จะเสนอซื้อให้ราคาสูงกว่านี้ ในราคาที่ John รับได้แล้วค่อยขาย หรือ John อาจจะเสนอขายเองที่ราคาที่สูงกว่านี้ก็ได้ เช่น

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

ผลลัพธ์ก็จะแสดงว่า

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

ซึ่งมันอาจจะเป็นอย่างนี้ตราบนานเท่านานก็ได้ ตราบใดที่ยังไม่มีคนเสนอราคาซื้อที่สูงกว่า 150 บาท

งั้นเราสมมุติเพิ่มเติมอีกหน่อยก็แล้วกัน ให้ John อยู่ในสถานะร้อนเงิน หุ้น KBANK เป็นหุ้นที่ไม่ดีในเศรษฐกิจแบบนี้ (ลูกหนี้ชักดาบเยอะ ค่าธรรมเนียมธนาคารก็เก็บไม่ได้) John จึงรีบตัดสินใจขาย แต่ขายแค่ 6 หุ้น และจะขายในราคาที่ไม่ต่ำกว่า 85 บาทต่อหุ้น John จึงส่งคำสั่งดังต่อไปนี้

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

ผลลัพธ์จึงเป็นดังนี้

```
-----------------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock KBANK are transferred from 'John' to 'Timothy'
Also, 200.0 baht is transferred from 'Timothy' to 'John'
-----------------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock KBANK are transferred from 'John' to 'Peter'
Also, 180.0 baht is transferred from 'Peter' to 'John'
-----------------------------------------------------------------------------------------
```

```
Matched!!! Now 2 shares of Stock KBANK are transferred from 'John' to 'Paul'
Also, 180.0 baht is transferred from 'Paul' to 'John'
```

เนื่องจากมีผู้มารอซื้อจำนวนมาก จึงเกิดคำสั่ง Match ขึ้นมากมาย ในกรณีนี้ John เสนอขายในราคาค่อนข้างต่ำคือ 85 บาทต่อหุ้น จึงเกิดการ Match ทั้งหมด (แต่คนที่มารอซื้อ ไม่ได้ทั้งหมด เพราะจำนวนหุ้นที่เสนอขายมีน้อย) Timothy ได้เสนอซื้อในราคาที่สูงที่สุด คือ 100 บาทต่อหุ้น จึงได้สิทธิ์ในการซื้อก่อนและมีการโอนเงินไป 2 x 100 = 200 บาท, คนต่อมาคือ Peter ซึ่งเสนอราคาซื้อไว้ที่ 90 บาทต่อหุ้น (และส่งคำสั่งเสนอมาก่อนคู่แข่งคือ Paul) จึงขายออกได้ 2 หุ้น โดยแลกกับเงิน 2 x 90 = 180 บาท ส่วน Paul เสนอซื้อมาที่ 90 บาทเหมือนกัน แต่ช้ากว่า จึงได้ทีหลัง ถึงแม้ว่า Paul จะเสนอซื้อ 3 หุ้นแต่ก็ได้ไปแค่ 2 หุ้นเพราะ John เสนอขายแค่ 6 หุ้น (ให้ไปกับคนก่อนหน้าแล้ว 4 จึงเหลือขายให้ Paul แค่ 2 หุ้น) เพราะฉะนั้นคำสั่งที่ค้างเหลืออยู่ในระบบตอนนี้จึงเป็น คำสั่งซื้อของ Paul 1 หุ้นที่ราคา 90 บาท, คำสั่งซื้อของ Titus 2 หุ้นที่ราคา 80 บาท และสุดท้ายคำสั่งขายของ John เอง 2 หุ้นแต่แรก ที่เสนอขายไว้ 150 บาท ถึงป่านนี้ก็ยังไม่มีใครจะซื้อที่ราคานี้ หากตรวจสอบ Portfolio ของ John จะพบว่ามีรายละเอียดดังนี้

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

```
-------------------------------------------
    Portfolio of 'John', 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 John มีหุ้น KBANK ลดลงจาก 10 หุ้นเหลือ 4 หุ้น (เพราะขายไปเมื่อกี้ 6 หุ้น) สังเกตว่าสินทรัพย์โดยรวม (Total portfolio values) มีมูลค่าลดลง เนื่องจากราคาหุ้นตก จากที่ซื้อมา 170 บาทต่อหุ้น ตอนนี้เหลือราคาที่ซื้อขายกันที่ 85 บาทต่อหุ้น เงินจึงหายวับจากที่ราคาหุ้นผันผวน อย่างไรก็ตาม John ก็จะบอกกับเพื่อน ๆ ว่า แค่นี้สิว 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("Matthew", "PTT", 3, 350);
market.submitSellOrder("Matthew", "PTT", 3, 360);
market.submitSellOrder("Matthew", "PTT", 4, 370);
market.submitSellOrder("Luke", "PTT", 5, 355);
market.submitSellOrder("Luke", "PTT", 5, 365);
market.submitSellOrder("Luke", "PTT", 5, 375);
market.submitSellOrder("James", "PTT", 3, 360);
market.submitSellOrder("James", "PTT", 3, 370);
market.submitSellOrder("James", "PTT", 4, 380);

market.submitBuyOrder("Mark", "PTT", 6, 370);
market.submitBuyOrder("John", "PTT", 6, 370);
market.submitBuyOrder("Peter", "PTT", 10, 370);
market.submitBuyOrder("Jude", "PTT", 10, 370);

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

```
market.submitBuyOrder("Paul", "PTT", 2, 270);

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

market.submitSellOrder("Titus", "PTT", 3, 320);
market.submitSellOrder("Titus", "PTT", 3, 240);
market.submitSellOrder("Titus", "PTT", 4, 220);
market.submitSellOrder("Mark", "PTT", 3, 320);
market.submitSellOrder("Mark", "PTT", 3, 240);
market.submitSellOrder("Mark", "PTT", 4, 220);

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

ผลลัพธ์

```
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Matthew' to 'Mark'
Also, 1050.0 baht is transferred from 'Mark' to 'Matthew'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Luke' to 'Mark'
Also, 1065.0 baht is transferred from 'Mark' to 'Luke'
--------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock PTT are transferred from 'Luke' to 'John'
Also, 710.0 baht is transferred from 'John' to 'Luke'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Matthew' to 'John'
Also, 1080.0 baht is transferred from 'John' to 'Matthew'
--------------------------------------------------------------------------------
Matched!!! Now 1 shares of Stock PTT are transferred from 'James' to 'John'
Also, 360.0 baht is transferred from 'John' to 'James'
--------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock PTT are transferred from 'James' to 'Peter'
Also, 720.0 baht is transferred from 'Peter' to 'James'
```

```
--------------------------------------------------------------------------------
Matched!!! Now 5 shares of Stock PTT are transferred from 'Luke' to 'Peter'
Also, 1825.0 baht is transferred from 'Peter' to 'Luke'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Matthew' to 'Peter'
Also, 1110.0 baht is transferred from 'Peter' to 'Matthew'
--------------------------------------------------------------------------------
Matched!!! Now 1 shares of Stock PTT are transferred from 'Matthew' to 'Jude'
Also, 370.0 baht is transferred from 'Jude' to 'Matthew'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'James' to 'Jude'
Also, 1110.0 baht is transferred from 'Jude' to 'James'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Titus' to 'Jude'
Also, 1110.0 baht is transferred from 'Jude' to 'Titus'
--------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Titus' to 'Jude'
Also, 1110.0 baht is transferred from 'Jude' to 'Titus'
--------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock PTT are transferred from 'Titus' to 'Paul'
Also, 660.0 baht is transferred from 'Paul' to 'Titus'
--------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock PTT are transferred from 'Titus' to 'Timothy'
Also, 660.0 baht is transferred from 'Timothy' to 'Titus'
--------------------------------------------------------------------------------
Matched!!! Now 1 shares of Stock PTT are transferred from 'Mark' to 'Timothy'
Also, 330.0 baht is transferred from 'Timothy' to 'Mark'
--------------------------------------------------------------------------------
Matched!!! Now 2 shares of Stock PTT are transferred from 'Mark' to 'Paul'
Also, 600.0 baht is transferred from 'Paul' to 'Mark'
--------------------------------------------------------------------------------
Matched!!! Now 1 shares of Stock PTT are transferred from 'Mark' to 'Paul'
Also, 270.0 baht is transferred from 'Paul' to 'Mark'
--------------------------------------------------------------------------------
```

```
Matched!!! Now 1 shares of Stock PTT are transferred from 'Mark' to 'Paul'
Also, 270.0 baht is transferred from 'Paul' to 'Mark'
-------------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Mark' to 'Timothy'
Also, 750.0 baht is transferred from 'Timothy' to 'Mark'
Stock 'PTT':        [Best Bid = 3x200.0 baht,  Best Offer = 2x320.0 baht]
```

เมื่อตรวจดูการกระจายหุ้น

```
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("Mark");
```

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

14. สมมุติว่าจากเหตุการณ์ข้อ 13 มีการซื้อขายกับอีกเล็กน้อยดังนี้

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

market.submitSellOrder("Peter", "PTT", 10, 125);
market.submitSellOrder("Jude", "PTT", 10, 75);
market.submitSellOrder("Paul", "PTT", 15, 75);
market.submitSellOrder("Timothy", "PTT", 15, 75);

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

ให้ผลลัพธ์คือ

```
-----------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Peter' to 'Timothy'
Also, 600.0 baht is transferred from 'Timothy' to 'Peter'
-----------------------------------------------------------------------------------
Matched!!! Now 7 shares of Stock PTT are transferred from 'Peter' to 'Matthew'
Also, 1050.0 baht is transferred from 'Matthew' to 'Peter'
-----------------------------------------------------------------------------------
Matched!!! Now 10 shares of Stock PTT are transferred from 'Jude' to 'Matthew'
Also, 1500.0 baht is transferred from 'Matthew' to 'Jude'
```

```
-----------------------------------------------------------------------------------------
Matched!!! Now 3 shares of Stock PTT are transferred from 'Paul' to 'Matthew'
Also, 450.0 baht is transferred from 'Matthew' to 'Paul'
-----------------------------------------------------------------------------------------
Matched!!! Now 12 shares of Stock PTT are transferred from 'Paul' to 'Matthew'
Also, 1200.0 baht is transferred from 'Matthew' to 'Paul'
-----------------------------------------------------------------------------------------
Matched!!! Now 8 shares of Stock PTT are transferred from 'Timothy' to 'Matthew'
Also, 800.0 baht is transferred from 'Matthew' to 'Timothy'
Stock 'PTT':        [Best Bid = 10x50.0 baht,  Best Offer = 7x75.0 baht]
```

คุณสังเกตได้หรือไม่ว่า Mark ไม่ได้เกี่ยวข้องกับการซื้อขายกันในข้อ 14. เลย คราวนี้มาดู Portfolio ของ Mark กัน

```
market.portfolio("Mark");
-------------------------------------------
    Portfolio of 'Mark', 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 Mark ลดลงอีก เหลือ 4575 เงินหายไป 10000 – 4575 = 5425 หรือ -54% นักศึกษาทราบหรือไม่ว่า เงิน Mark หายไปไหน ทั้ง ๆ ที่ไม่ได้เกี่ยวข้องกับการซื้อขายหุ้นเลย

- การบ้านนี้อาจารย์ได้ implement ให้เกือบเสร็จหมดแล้วครับ เหลือเพียงแค่สองคลาสสุดท้ายเท่านั้น คือ **class Node และ class Heap ที่คุณต้องแก้ไขโค้ดเพิ่มเติมจนเสร็จ** และให้ทำงานได้ดังโจทย์ด้านบน

ภาคผนวก

Test Case ดังต่อไปนี้ จะช่วยให้นักศึกษาเข้าใจการทำงานของ 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 มีราคาที่ 30 บาท เข้าคิวมาตอน Timestamp เป็น 1 ส่วน Node B มีราคาที่ 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

```java
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));
}
```

Output

```
nodeA.compare(nodeB) = false
nodeB.compare(nodeA) = true
nodeA.compare(nodeB) = true
nodeB.compare(nodeA) = false
nodeA.compare(nodeB) = true
nodeB.compare(nodeA) = false
```