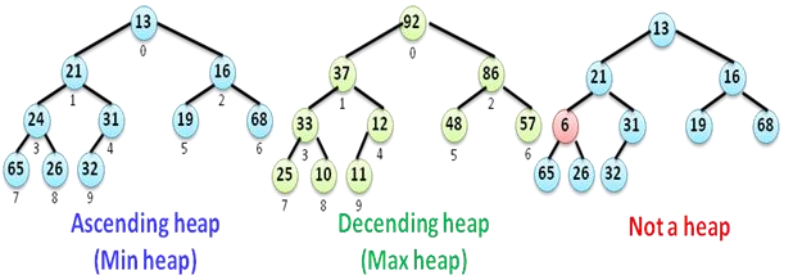
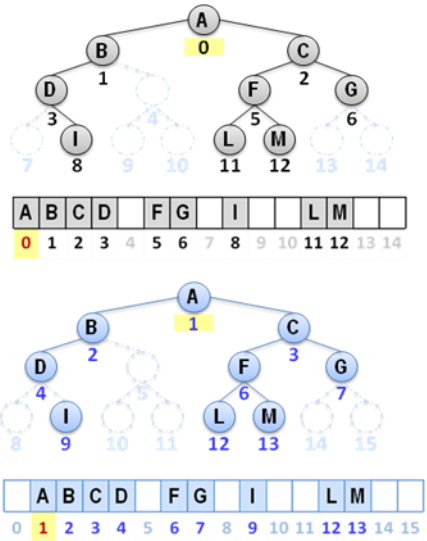
Lab 10 : Heap & Lab 11 : Sort

วัตถุประสงค์ ฝึกหัดเขียน Sort แบบต่างๆ



Binary Heap : complete Binary Tree ซึ่ง key ของ node ใดๆ
1. <= key ของ decendents ของ มัน : **Min heap**
   เช่น 13 น้อยกว่าลูกหลานทั้งหมดของมัน
2. >= key ของ decendents ของ มัน : **Max heap**
   เช่น 37 มากกว่าลูกหลานทั้งหมดของมัน

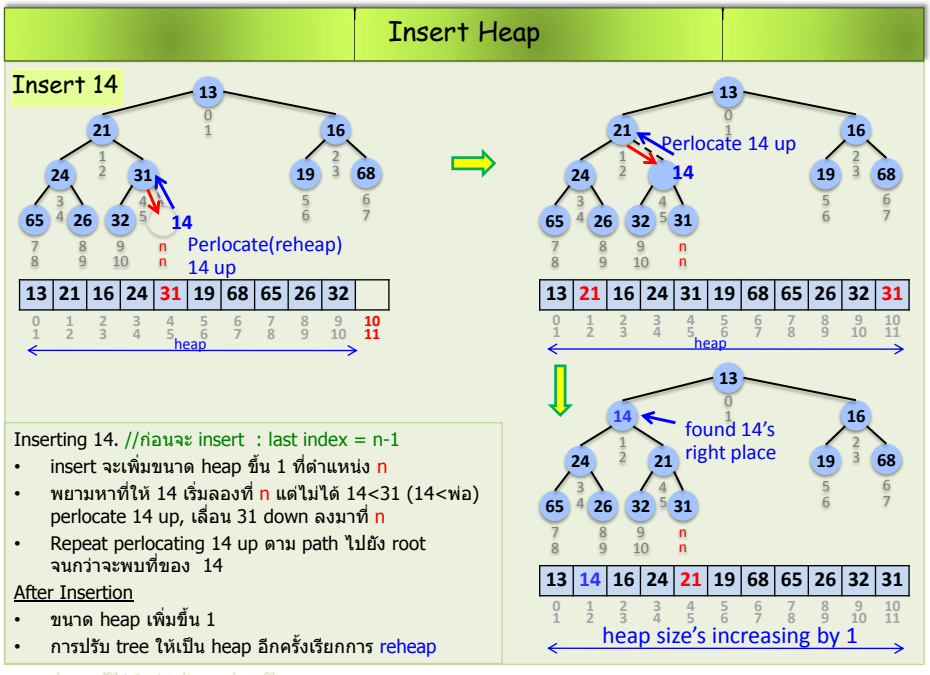
เนื่องจากเป็น complete binary tree จึงควร implement ด้วย data structure แบบ _____



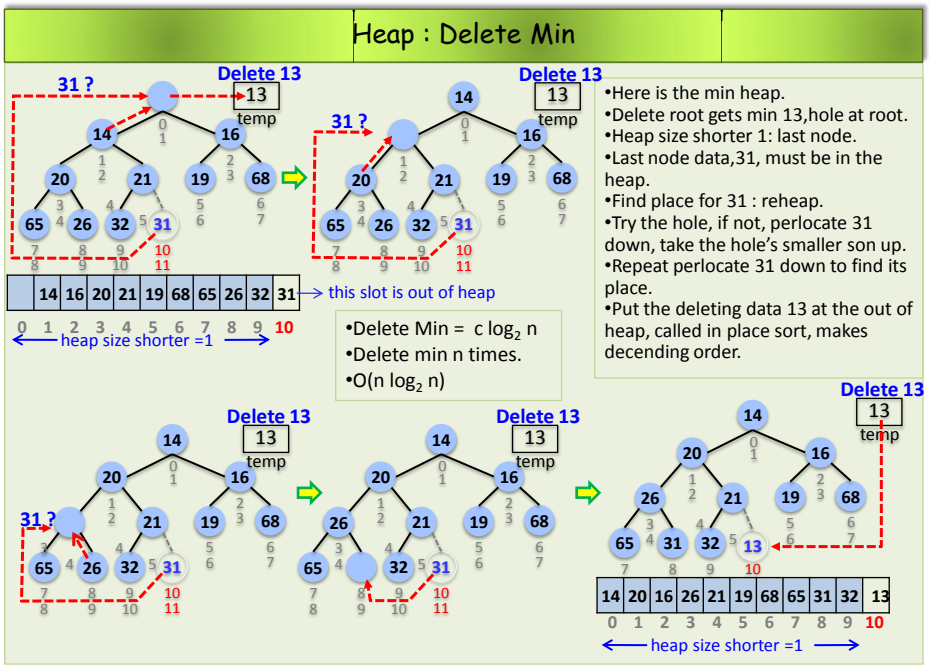
ซึ่งโครงสร้างแบบ implicit array (sequential array) นี้ เราจะนับตำแหน่งของ node นิยมทำกับ 2 แบบ คือ ให้ root เริ่มที่ 0 (รูปซ้าย) และ ให้ root เริ่มที่ 1 (รูปขวา) นับตำแหน่งไปเรื่อยๆ เรียงตามลำดับทีละ level จากซ้ายไปขวา นับไม่เว้น node ที่ไม่มี (ในรูปเป็นเส้นประ) แต่ละ node จึงมีตำแหน่งกำกับดังรูป และใช้ implicit array (sequential array) เก็บข้อมูล data ของแต่ละ node ใน index ตามตำแหน่งของมัน

จะเห็นว่า data structure แบบ implicit array นี้ เก็บเฉพาะ data ไม่จำเป็นต้องเก็บ link เนื่องจากสามารถคำนวณได้

	เริ่ม root ที่ index 0	เริ่ม root ที่ index 1
ลูกข้างซ้ายของ node ที่ index i อยู่ที่ index	_____	_____
,, ขวา ,, ,,	_____	_____



Algorithm ในการสร้าง heap ใช้วิธี insert data เข้าไปใน heap ทีละตัว ซึ่งทำให้ size ของ heap เพิ่มขึ้นจาก index สุดท้ายไป 1 ตัว ในรูปกำลัง insert 14 index ที่เพิ่มขึ้น n คือ 10 (root เริ่มที่ index 0) / 11 (root เริ่มที่ index 1) โดยดูว่าสามารถ insert data เข้าที่ตำแหน่ง n ได้หรือไม่ (ผิดกฎของ heap หรือไม่) หากไม่ได้ จะทำการ perlocate data up ขึ้นไปตามทางพ่อของมันไปยัง root ทำให้การ insert data 1 ตัวทำงานอย่างมากเท่ากับความสูงของ tree คือ $\log_2 n$ ดังนั้น insert n ตัว มีลำดับเป็น _____



min heap ข้อมูลตัวที่น้อยที่สุดอยู่ที่ _____ การ delete min คือการดึง root ออกมา และทำการ reheap (จัดข้อมูลที่เหลือให้เป็น heap ใหม่) ข้อมูลที่น้อยเป็นลำดับถัดมาจะขึ้นไปอยู่ที่ root ดังนั้น หากวน delete min ออกมาเรื่อยๆ จนหมด heap เราจะได้ข้อมูลที่ sorted จากน้อยไปมาก ascending order การทำ in place sort คือ delete min แล้วนำมาใส่ไว้ใน array ตัวเดิม โดยใส่ไว้ที่ index ตัวสุดท้ายของ heap ซึ่งจะเป็นตำแหน่งที่หายจาก heap เมื่อ delete data ออกหาก delete min ออกจนหมด การทำ in place sort บน min heap จะให้ array ที่เรียงแบบ _____ order จากมากไปน้อย

การทดลอง 1 Heap Sort (in place) เขียนฟังก์ชั่นต่อไปนี้โดยคิด parameter เอง พร้อมทั้งทดสอบความถูกต้องด้วย

1) เพื่อทดสอบความถูกต้องของ heap เขียนฟังก์ชั่นเพื่อพิมพ์ tree
   - พิมพ์ list l
   - print90 ( ) พิมพ์รูป tree ของ list l ในรูปหมุนซ้าย 90 องศา

   กำหนด input list l พร้อมรันฟังก์ชั่นในข้อ 1 เพื่อทดสอบ

```
l = [68,65,32,24,26,21,19,13,16,14]
```

```
68  65  32  24  26  21

    19
  32
    21
68
    26
      14
  65
      16
    24
      13
```

2) กำหนด heap `h = []` เขียนฟังก์ชั่น insert( ) เพื่อ insert data เข้าไปใน heap h ให้วน loop insert data จาก list l เข้าไปใน heap h ทีละตัวจนหมด แต่ละครั้งที่ insert ให้พิมพ์ heap h เพื่อทดสอบ

```
insert 68
68
68
-----------
insert 65
65 68
65
   68
-----------
insert 32
32 68 65
   65
32
   68
-----------
insert 24
24 32 65 68
   65
24
   32
      68
```

```
insert 26
24  26  65  68  32

  65
24
    32
  26
    68
---------
insert 21
21  26  24  68  32  65

  24
    65
21
    32
  26
    68
---------
insert 19
19  26  21  68  32  65  24

    24
  21
    65
19
    32
  26
```

```
insert 13
13  19  21  26  32  65  24  68

    24
  21
    65
13
    32
  19
    26
      68
---------
insert 16
13  16  21  19  32  65  24  68  26

    24
  21
    65
13
    32
  16
      26
    19
      68
---------
```

```
insert 14
13  14  21  19  16  65  24  68  26  32

    24
  21
    65
13
    16
      32
  14
      26
    19
      68
---------
```

3) ทำ Heap Sort โดยเขียนฟังก์ชั่น deleteMin() เพื่อทำการดึงค่า root ของ heap h ในตัวอย่างทำ inplace sort heap ปรับ reheap พร้อมทั้งลดค่า last index ของ heap ลง ทดสอบความถูกต้องโดยให้พิมพ์ heap ทั้งหมด วน loop deleteMin จนหมดทุกตัวมาใส่ที่ list a พิมพ์ a จะได้ ascending list พิมพ์ แต่ถ้าทำ inplace sort พิมพ์ h จะได้ decending list ข้างล่างเป็นตัวอย่าง output

```
***deleteMin***
input heap:
13 14 21 19 16 65 24 68 26 32
      24
   21
      65
13
      16
         32
   14
         26
      19
         68
==== heap sort ====
deleteMin =13 FindPlaceFor 32
14 16 21 19 32 65 24 68 26 13
      24
   21
      65
14
      32
         13
   16
         26
      19
         68
```

```
deleteMin =14 FindPlaceFor 26
16 19 21 26 32 65 24 68 14 13
      24
   21
      65
16
      32
         13
   19
         14
      26
         68
deleteMin =16 FindPlaceFor 68
19 26 21 68 32 65 24 16 14 13
      24
   21
      65
19
      32
         13
   26
         14
      68
         16
```

```
deleteMin =19 FindPlaceFor 24
21 26 24 68 32 65 19 16 14 13
      19
   24
      65
21
      32
         13
   26
         14
      68
         16
deleteMin =21 FindPlaceFor 65
24 26 65 68 32 21 19 16 14 13
      19
   65
      21
24
      32
         13
   26
         14
      68
         16
```

```
deleteMin =24 FindPlaceFor 32
26 32 65 68 24 21 19 16 14 13
      19
   65
      21
26
      24
         13
   32
         14
      68
         16
deleteMin =26 FindPlaceFor 68
32 68 65 26 24 21 19 16 14 13
      19
   65
      21
32
      24
         13
   68
         14
      26
         16
```

```
deleteMin =32 FindPlaceFor 65
65 68 32 26 24 21 19 16 14 13
      19
   32
      21
65
      24
         13
   68
         14
      26
         16
deleteMin =65 FindPlaceFor 68
68 65 32 26 24 21 19 16 14 13
      19
   32
      21
68
      24
         13
   65
         14
      26
         16
==== Sorting a ====
13 14 16 19 21 24 26 32 65 68
```

การทดลอง 2 เขียน Sorting แบบต่างๆ ตาม algorithm ที่เรียน ทดลองโดยกำหนด data เองให้ครอบคลุมทุกกรณี หากทำไม่ได้ ดู Pseudocode คิดตาม algorithm ถึง complexity ของแต่ละ sort ตามที่เรียน ลองใส่ code print จำนวนการ compare data ในแต่ละ pass ในกรณีต่างๆ best case และ worst case หากทำไม่ได้ดูเฉลย code ในภาคทฤษฎี