Homework 3

1. a. The minimum distance between two cloest numbers is $|X-Y|$

Algorithm distance between two Points (P)

// Input : list P of $n \geq 2$ Points $P_1 = (x_1, y_2)$ ... $P_n(x_n, y_n)$

// output : Indicate the Closest distance between Index1 and Index2

minimum distance = ∞

new vector

dmin = ∞

for i=1 to n-1

for j=i+1 to n

distance = $sqrt((x_i - x_j)^2 + (y_i - y_j)^2)$

push_back distance into new vector

return new vector.

Algorithm Presort (distance between array or vector, integer)

// Input array or vector of distance between two points

// output true or false the integer whether in the array.

merge_sort (distancebetween array [0,... n-1]

2, a, Brute force Intersection (A, B)

\\ input two different sets . A and B, A has n umbers, B has m numbers

\\ output one new set of insection A and B

```
for i = 0 to n-1
    for j = 0 to m - 1
    if ( A[i] = B[j] )
        push_back A[i] to new set C
return C
```

Time running in Brute force Algorithm for Intersection is

$T(n) = m \times n = m \times n$

b. Presorting - Based Algorithm

Sort set A and set B first.

```
Merge_Sort ( A {0, ... n-1} )
Merge- Sort ( B {0, ... m-1} )
    i = 0
    j = 0
  while ( i < m and j < n )
    if( A[i] = B[j] )
        ++i
        ++j
      push_ back A[i] to C
    else if ( A[i] > B[j])
        ++j
    else if (A[i] < B[j])
        ++i
  return C
```

Time running for the Presort Based Algorithm is

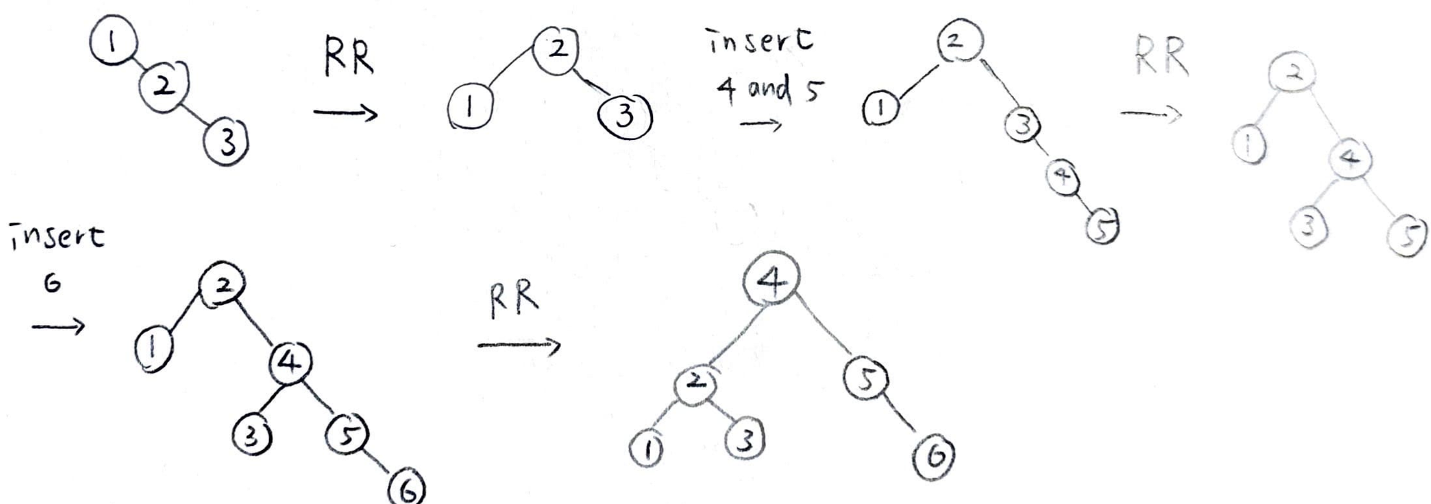
$T(n) = \log m + \log n + m + n$

3. $$\begin{array}{c} R1 \\ R2 \\ R3 \end{array}\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 3 \\ 1 & -1 & 3 & 8 \end{array}\right] \xrightarrow{-R1+R3} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 3 \\ 0 & -2 & 2 & 6 \end{array}\right] \xrightarrow{-2R_1+R_2} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & -1 & -1 & -1 \\ 0 & -2 & 2 & 6 \end{array}\right]$$

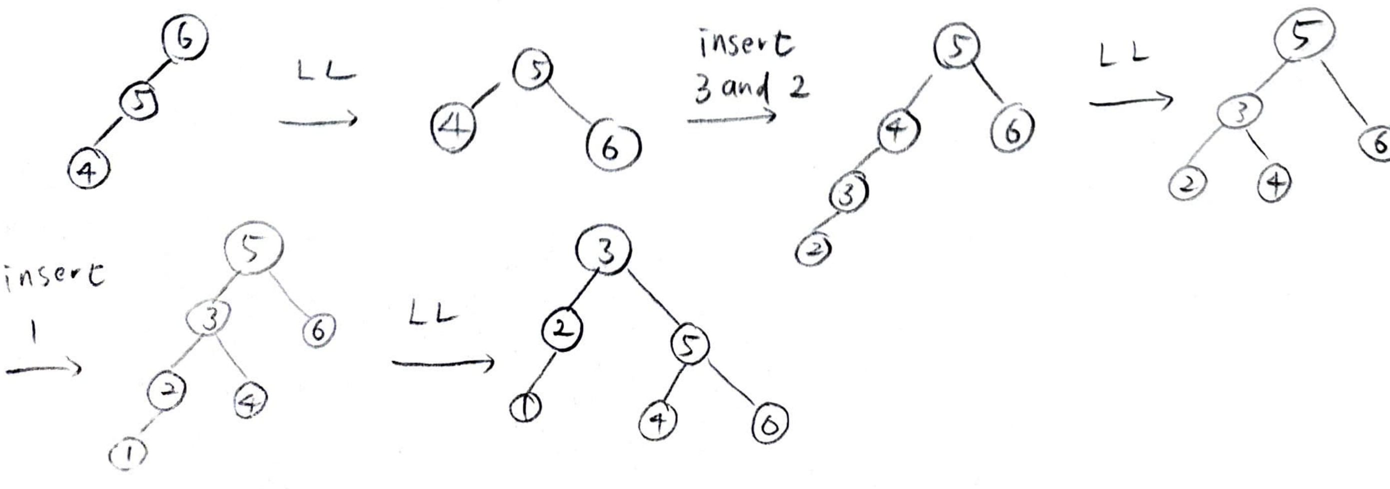
$$\xrightarrow{-R_2+R_3} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & -1 & -1 & -1 \\ 0 & 0 & 4 & 8 \end{array}\right] \xrightarrow{R3\times\frac{1}{4}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 2 \end{array}\right] \xrightarrow{R_2\times-1\,+\,R_3\times-1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

$$\xrightarrow{R_1+R_2\times-1} \left[\begin{array}{ccc|c} 1 & 0 & 1 & 3 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \end{array}\right] \xrightarrow{R_1+R_3\times-1} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \end{array}\right]$$ #

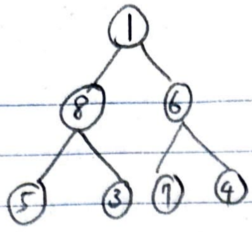
4. a. 1 2 3 4 5 6



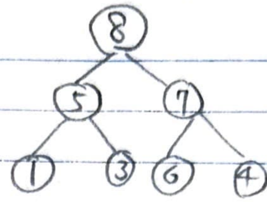
b. 6, 5, 4 3. 2, 1



5. a. array = {1, 8, 6, 5, 3, 7, 4}

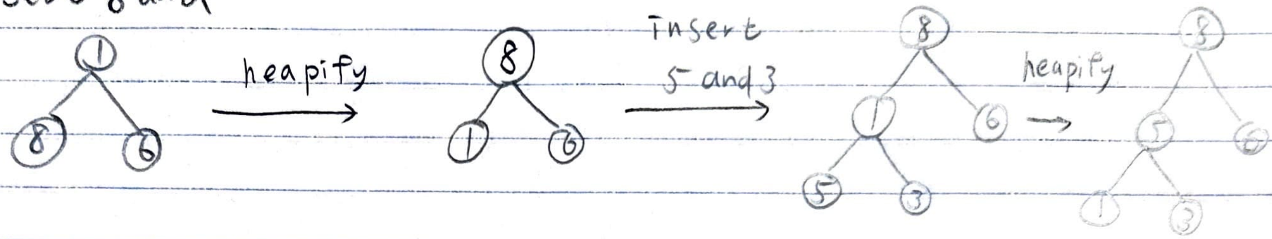


max heap

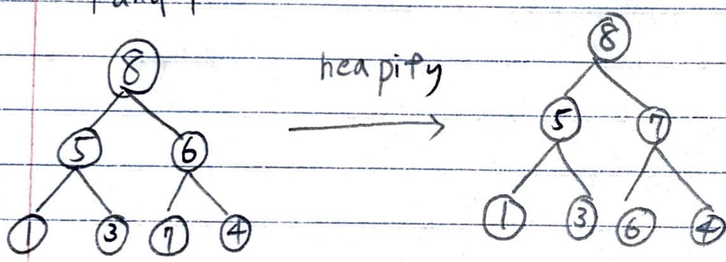


b. for max heap.

insert 8 and



Insert
7 and 4



C. Yes, it is always true for the two different input
Because the index for each item is the same.
it won't affect the output of heap.