Student Name: Duc Tu Luong
Last 3-digit ID: 122

Homework #4

Part A

1. Solved exercise #1 in Chapter 5

a. Pseudo code

$Let\ A\ be\ the\ set\ with\ n\ entries$
$Let\ i\ be\ the\ index\ of\ the\ first\ element\ and\ let\ i = 0\ initially$
$Let\ j\ be\ the\ index\ of\ the\ last\ element\ and\ let\ j = n\ initially$
$Let\ p\ be\ the\ index\ of\ peak\ element$

$FindPeak(i, j)\ \{$
$\quad p = round\ of\ \frac{(i+j)}{2}$
$\quad If\ (A[p-1] < A[p] < A[p+1])$
$\quad\quad Let\ i = p$
$\quad\quad FindPeak(i, j)$
$\quad Endif$
$\quad If\ (A[p-1] > A[p] > A[p+1])$
$\quad\quad Let\ j = p$
$\quad\quad FindPeak(i, j)$
$\quad Endif$
$\quad If\ (A[p-1] < A[p]\ \&\&\ A[p] > A[p+1])$
$\quad\quad return\ p$
$\quad Endif$
$\}$

b. Problem instance of size 10

$Let\ A = \{2, 3, 6, 8, 9, 10, 11, 7, 5, 4\}$
$Let\ i = 0$
$Let\ j = 10$

$-\ Step\ 1: FindPeak(0, 10)$
$$p = \frac{0+10}{2} = 5$$
$A[p-1] = A[5-1] = A[4] = 8$
$A[p] = A[5] = 9$
$A[p+1] = A[5+1] = A[6] = 10$
$\rightarrow\ A[p-1] < A[p] < A[p+1]$
$\rightarrow\ i = p = 5$

$- Step\ 2: FindPeak(5, 10)$

$$p = \frac{5+10}{2} = 7.5 = 8\ (round\ to\ 8)$$
$$A[p-1] = A[8-1] = A[7] = 11$$
$$A[p] = A[8] = 7$$
$$A[p+1] = A[8+1] = A[9] = 5$$
$$\rightarrow \quad A[p-1] > A[p] > A[p+1]$$
$$\rightarrow \quad j = p = 8$$

$- Step\ 3: FindPeak(5, 8)$

$$p = \frac{5+8}{2} = 6.5 = 7\ (round\ to\ 7)$$
$$A[p-1] = A[7-1] = A[6] = 10$$
$$A[p] = A[7] = 11$$
$$A[p+1] = A[7+1] = A[8] = 7$$
$$\rightarrow \quad A[p] > A[p-1]\ and\ A[p] > A[p+1]$$
$$\rightarrow \quad return\ p = 7$$

c. Time complexity: $O(\log n)$

2. Solved exercise #2 in Chapter 5

a. Pseudo code

$Let\ S\ be\ the\ array\ containing\ days\ with\ fixed\ price\ for\ each\ day$
$Find - Opt(S)$
  $If\ S\ is\ empty$
    $return$ empty
  $If\ S\ has\ 1\ element$
    $return\ \big((S[0], S[0]), S[0], S[0]\big)$
  $Else$
    $Divide\ the\ list\ S\ into\ 2\ halves$
      $A\ contains\ the\ first\ \frac{n}{2}\ elements$
      $B\ contains\ the\ remaining\ \frac{n}{2}\ elements$
    $Let\ minA, maxA\ be\ the\ minimum\ and\ maximum\ values\ of\ A$
    $Let\ minB, maxB\ be\ the\ minimum\ and\ maximum\ values\ of\ B$
    $Let\ optA\ and\ optB\ be\ the\ optimal\ solution\ for\ A\ and\ B$
    $(optA, minA, maxA) = Find - Opt(A)$
    $(optB, minB, maxB) = Find - Opt(B)$
    $min = \min(minA, minB)$
    $\max = \max(maxA, maxB)$
    $If\ maxB > minA$
      $optAB = (minA, maxB)$
      $opt = \max(optA, optB, optAB)$
    $Else$

$opt = \max(optA, optB)$
$return\ (opt, min, max)$
$Endif$

b. Problem instance of size 10
$Let\ S = [3, 7, 6, 2, 1, 9, 5, 10, 4, 8]$

Divide into: A = [3, 7, 6, 2, 1] and B = [9, 5, 10, 4, 8]
1st half: A = [3, 7, 6, 2, 1] divides into [3, 7] and [6, 2, 1]

1. $Find - Opt([3, 7])$
   $optA = (3, 3), minA = 3, maxA = 3$
   $optB = (7, 7), minB = 7, maxB = 7$
   $max = \max(maxA, maxB) = \max(3, 7) = 7$
   $min = \min(minA, minB) = \min(3, 7) = 3$
   $maxB > minA \rightarrow optAB = (minA, maxB) = (3, 7)$
   $opt = \max(optA, optB, optAB) = optAB = (7, 3)$
   $return\ (opt, min, max) = ((3, 7), 3, 7)$

2. $Find - Opt([6, 2])$
   $optA = (6, 6), minA = 6, maxA = 6$
   $optB = (2, 2), minB = 2, maxB = 2$
   $max = \max(maxA, maxB) = \max(6, 2) = 6$
   $min = \min(minA, minB) = \min(6, 2) = 2$
   $opt = \max(optA, optB) = (6, 6)$
   $return\ (opt, min, max) = ((6, 6), 2, 6)$

3. $Find - Opt([6, 2, 1])$
   $optA = (6, 6), minA = 2, maxA = 6$
   $optB = (1, 1), minB = 1, maxB = 1$
   $max = \max(maxA, maxB) = \max(6, 1) = 6$
   $min = \min(minA, minB) = \min(2, 1) = 1$
   $opt = \max(optA, optB) = (6, 6)$
   $return\ (opt, min, max) = ((6, 6), 1, 6)$

4. $Find - Opt([3, 7, 6, 2, 1])$
   $optA = (3, 7), minA = 3, maxA = 7$
   $optB = (6, 6), minB = 1, maxB = 6$
   $max = \max(maxA, maxB) = \max(7, 6) = 7$
   $min = \min(minA, minB) = \min(3, 1) = 1$
   $maxB > minA \rightarrow optAB = (minA, maxB) = (3, 6)$
   $opt = \max(optA, optB, optAB) = (3, 7)$
   $return\ (opt, min, max) = ((3, 7), 1, 7)$

5. $Find - Opt([9, 5])$
   $optA = (9, 9), minA = 9, maxA = 9$

$optB = (5, 5), minB = 5, maxB = 5$
$max = \max(maxA, maxB) = \max(9, 5) = 9$
$min = \min(minA, minB) = \min(9, 5) = 5$
$opt = \max(optA, optB) = (9, 9)$
$return\ (opt, min, max) = ((9, 9), 5, 9)$

6. $Find - Opt([10, 4])$
$optA = (10, 10), minA = 10, maxA = 10$
$optB = (4, 4), minB = 4, maxB = 4$
$max = \max(maxA, maxB) = \max(10, 4) = 10$
$min = \min(minA, minB) = \min(10, 4) = 4$
$opt = \max(optA, optB) = (10, 10)$
$return\ (opt, min, max) = ((10, 10), 4, 10)$

7. $Find - Opt([10, 4, 8])$
$optA = (10, 10), minA = 4, maxA = 10$
$optB = (8, 8), minB = 8, maxB = 8$
$max = \max(maxA, maxB) = \max(10, 8) = 10$
$min = \min(minA, minB) = \min(4, 8) = 4$
$maxB > minA \rightarrow optAB = (minA, maxB) = (4, 8)$
$opt = \max(optA, optB, optAB) = (4, 8)$
$return\ (opt, min, max) = ((4, 8), 4, 10)$

8. $Find - Opt([9, 5, 10, 4, 8])$
$optA = (9, 9), minA = 5, maxA = 9$
$optB = (4, 8), minB = 4, maxB = 10$
$max = \max(maxA, maxB) = \max(9, 10) = 10$
$min = \min(minA, minB) = \min(5, 4) = 4$
$maxB > minA \rightarrow optAB = (minA, maxB) = (5, 10)$
$opt = \max(optA, optB, optAB) = (5, 10)$
$return\ (opt, min, max) = ((5, 10), 4, 10)$

9. $Find - Opt([3, 7, 6, 2, 1, 9, 5, 10, 4, 8])$
$optA = (3, 7), minA = 1, maxA = 7$
$optB = (5, 10), minB = 4, maxB = 10$
$max = \max(maxA, maxB) = \max(7, 10) = 10$
$min = \min(minA, minB) = \min(1, 4) = 1$
$maxB > minA \rightarrow optAB = (minA, maxB) = (1, 10)$
$opt = \max(optA, optB, optAB) = (1, 10)$
$return\ (opt, min, max) = ((1, 10), 1, 10)$

Result: buy on day 5 which has price of 1 and sell on day 8 which has price of 10

c. Time complexity: $O(n \log n)$

Part B

1. Chapter 5, Exercise 2 (Significant inversion problem)
   a. *Model of problem:* Given a sequence of n numbers $a_1, a_2, \dots a_n$. Let's call a pair a significant inversion if $i > j \text{ and } a_i > 2a_j$. Give an *O(n* log n) algorithm to count the number of significant inversions between two orderings.

   b. *Algorithm pseudo code*
   $Let\ inversion\ be\ the\ number\ of\ significant\ inversions$
   $Merge - and - Count(inversion, A, B)$
   $\quad Let\ i\ be\ the\ current\ pointer\ of\ A, i = first\ index\ of\ A\ initially$
   $\quad Let\ j\ be\ the\ current\ pointer\ of\ B, j = first\ index\ of\ B\ initially$
   $\quad Let\ merged\ be\ the\ merged\ list\ to\ return, merged\ is\ empty\ initially$
   $\quad While\ i < length\ of\ A\ and\ j < length\ of\ B$
   $\quad\quad If\ A[i] < B[j]$
   $\quad\quad\quad append\ A[i] to\ merged$
   $\quad\quad\quad i++$
   $\quad\quad Else\ if\ B[j] < A[i]$
   $\quad\quad\quad append\ B[j] to\ merged$
   $\quad\quad\quad If\ 2B[j] < A[i]$
   $\quad\quad\quad\quad inversion++$
   $\quad\quad\quad Endif$
   $\quad\quad\quad j++$
   $\quad\quad Endif$
   $\quad Endwhile$
   $\quad If\ i < length\ of\ A$
   $\quad\quad append\ all\ remaining\ elements\ in\ A\ to\ merged$
   $\quad Else$
   $\quad\quad append\ all\ remaining\ elements\ in\ B\ to\ merged$
   $\quad Endif$
   $\quad return\ (inversion, merged)$
   $Sort - and - Count(inversion, L)$
   $\quad If\ L\ has\ one\ element$
   $\quad\quad return\ (0, L)$
   $\quad Else$
   $\quad\quad Divide\ L\ into\ 2\ halves$
   $\quad\quad\quad A\ contains\ the\ first\ \left\lfloor \frac{n}{2} \right\rfloor\ elements$
   $\quad\quad\quad B\ contains\ the\ remaining\ \left\lceil \frac{n}{2} \right\rceil\ elements$
   $\quad\quad (r_A, A) = Sort - and - Count(inversion, A)$
   $\quad\quad (r_B, B) = Sort - and - Count(inversion, B)$
   $\quad\quad (r, L) = Merge - and - Count(inversion, A, B)$
   $\quad Endif$
   $\quad return\ (r_A + r_B + r, L)$

c. *Time complexity:* $O(n \log n)$

d. Implementation

$Input$: $[10, 1, 5, 7, 2, 8, 6, 4, 9, 3]$

$Output$:

$Significant\ inversions$: 9

$Sorted\ list$: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

2. Chapter 5, Exercise 6 (Local minimum problem)

a. *Model of problem*

Given an n-node complete binary tree T, where n = 2^d - 1 for some d. Each node v of T is labeled with a real number x_v. Assume that the real numbers labeling the nodes are all distinct. A node v of T is a *local minimum* if the label x_v is less than the label x_w for all nodes w that are joined to v by an edge.

Given such a complete binary tree T, for each node v, the value x_v is determined by *probing* the node v. Find a local minimum of T using only O(log *n) probes* to the nodes of T.

b. Algorithm pseudo code

$Let\ T\ be\ the\ n - node\ binary\ tree$

$Let\ V\ be\ the\ root\ vertex\ in\ T$

$Find - Local - Min(V)$

$\quad If\ V\ has\ children\ nodes$

$\quad\quad Let\ L\ and\ R\ be\ V's\ children$

$\quad\quad Probe\ the\ values\ of\ x_L, x_R, and\ x_V$

$\quad\quad If\ x_V == \min(x_L, x_R, x_V)$

$\quad\quad\quad return\ V$

$\quad\quad Else\ if\ x_L < x_V$

$\quad\quad\quad return\ Find - Local - Min(L)$

$\quad\quad Else\ if\ x_R < x_V$

$\quad\quad\quad return\ Find - Local - Min(R)$

$\quad\quad Endif$

$\quad Else$

$\quad\quad return\ V$

c. Time complexity: $O(\log n)$