[redacted] **Class no. ____________**

Question 1. [13 points. 1 point each] Select the best answer (or one correct answer).

1. This question is about the rotations we do on a Red-Black Trees.
   (1) It preserves the black height of all nodes.
   (2) It preserves “inorder” ordering.

   (A) (1) alone (B) (2) alone **(C) Both (1) and (2)** (D) Neither (1) nor (2).

2. An array sorted in the descending order.

   (A) is never a max heap (B) may or may not be a min heap

   (C) is always a max heap (D) is always a min heap

3. Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced.)?
   (A) Heap Sort (B) Quick Sort **(C) Insertion Sort** (D) Merge Sort

4. Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?
   (A) **Heap Sort with time complexity O(nlogk)**
   (B) Quick Sort with time complexity O(klogk)
   (C) Insertion Sort with time complexity O(kn)
   (D) Merge Sort with time complexity O(klogk)

5. If a Job’s priority is completely determined by its arrival time in the sense that any job that arrived at10:00 AM has less priority than any Job that arrives at 10:01 AM, which of the following is the best to use in this context as a priority queue?
   (A) Max heap (B) Queue (C) **Stack** (D) Min heap

6. If a child and the parent have the same black-height in a Red-Black tree, what is the color of the child?
   (B) Can be Red or Black. (B) Black **(C) Red** D) No such child is possible.

7. Consider a Red-Black tree with n internal nodes and height h. Then
   (A) $h = 2\log(n - 1)$ (B) $h \geq 2\log(n + 1)$ (C) **$h \leq 2\log(n + 1)$** (D) $h \leq 2\log(n - 1)$

8. Let G(V, E) be a connected graph such that |V| = 9. Select the statement that is always true

   (A) $|E| \geq 29$ **(B) $|E| \geq 8$** (C) $|E| \geq 36$ (D) $|E| \geq 9$

9. Let G be a graph. Let G^c be the complement of G. Select the statement that is always true.

   (A) G or G^c is connected.
   (B) **If G is connected, G^c is disconnected**.
   (C) If G is connected, G^c is connected.
   (D) All of the above.

10. In order to determine whether or not a graph has an odd cycle, we use

    (A) Topological ordering (B) BFS **(C) DFS** (D) none of the above

11. A certain university has the following policy: A faculty cannot be a student and a student cannot be a faculty. Suppose you created a graph G with vertex set as all faculty and students. If the purpose of the G is to represent the relationship between “teacher” and “student” (such as Mr. A teaches Mr. B) then the graph G is a

    (A) forest (B) complete graph (C) disconnected graph (D) **bipartite graph**

12. A graph G has exactly two components Then G is a

    (A) forest (B) complete graph (C) **disconnected graph** (D) bipartite graph

13. Let C be a connected component of a graph. If |V(C)| = k, then the maximum number of edges in C is

    (A) $k^2 / 2$.
    (B) $(k^2 + k)/2$.
    (C) k^2.
    (D) $(k^2 – k)/2$

9 correct 9 points.

Question 2. [5 points. 1/2 point each] True or False questions.

(a) There is a graph with seven vertices such that its vertices has the following vertex degrees: 3, 5, 3, 4, 2, 6, 4. **WRONG**

(b) A connected component on n vertices and n edges may or may not have a cycle. **CORRECT**

(c) Every bipartite graph must have two connected components.**WRONG**

(d) There is no Red-Black tree such that number of Red nodes is twice the number of internal Black nodes.**CORRECT** from WRONG

(e) We use dynamic programming to reduce the complexity due to the dynamic nature of the problem.**WRONG**

(f) In the case of an undirected graph, a minimum spanning tree can be used to compute the shortest path between any two vertices.**CORRECT**

(g) Not every NP-Complete problem can be verified in Polynomial time.**CORRECT**

(h) There are problems in P that are not in NP. **CORRECT**

(i) If there is a polynomial reduction from problem A to Problem B means if you have polynomial time algorithm to solve Problem A, then we have a polynomial time algorithm for Problem B.**CORRECT**

(j) A problem P is ***NP**-hard* if for *every* problem S in ***NP***, P is polynomial reducible to S.**CORRECT**

7 correct answers. That is 3.5 points.

Question 3. [3 points] You are placing numbers 1, 2, 3, 4 and 5 on the five nodes of a complete binary tree. Please answer all questions (a) – (c) based on these facts. (Remember, a complete binary tree can be shown in the array form (as we did in the case of heap), No drawing required.)

a. Please show **one complete binary** tree (with numbers 1, 2, 3, 4 and 5) that satisfies all three conditions: is **not** a BST, is **not** a max-heap and is **not** a min-heap.
b. Please show **all complete binary trees** (with numbers 1, 2, 3, 4 and 5) **that is a BST.**
c. Please show **all** complete binary trees (with numbers 1, 2, 3, 4 and 5) that are Red-Black Trees. Please remember to indicate the color of each node.

a) |3|5|1|2|4   CORRECT   1 Point

b) 4|2|5|1|3   CORRECT   0.5 Point   There are two answers.

c) there are 2 arrays: CORRECT 1 Point

|4B 2R 5B 1B 3B|

|4B 2B 5B 1R 3R|

[redacted]

Question 4. [3 Points] (to be illustrated step by step in detail. Without details you will get 1 Point for the correct answer.)

Compute the shortest path from A to F. **Show all steps**.

For this problem alone, we will use the notation <X, Y , k> to mean there is a **directed edge** from X to Y with weight k.

Graph is [<A, B, 4>, <A, C, 5>, <B, C, -1>, <B, D, -4>, <C, E, -3>, <D, C, 3>, <D, E, 6>,

<D, F, 5>, <E, F, 9>]

Your answer

Shortespath is: G= <A,B,4>, <B,D, -4>,< D,F,5> = **5**

[redacted]

[redacted]

Question 5. [3 Points] (to be illustrated step by step in detail. Without details you will get 1 Points for the correct answer.)

Compute minimum spanning tree.

For this problem alone, we will use the notation (X, Y, k) to mean there is an **undirected edge** between X and Y with weight k.

Graph is = [(S, A, 3), (S, B, 3), (A, D, 5). (A, C, 4), (B, C, 2), (B, F, 3), (C, D, 7), (C, E, 6), (C, F, 8), (D, G, 2), (E, G, 2), (E, H, 3), (F, H, 3), (G, T, 6), (H, T, 1)

Your answer

Minimum Spanning tree is =

1- S A B C D E F G H T

2- S A B C D E F G **HT**

3 - S A D E F G **BC HT**

4- S A E F **DG BC HT**

5- S A F **DEG BC HT**

6- F **SA BC DEG BC HT**

7- F **SABC DEG HT**

8- **SABCF DEG HT**

[redacted]

[redacted]

Question 6. [3 Points]

(1) **[1 point]** Show the **max heap** that is built using **Top-Down** Algorithm on the data 10, 20, 30, 5, 45, 70, 80.
(2) **[1 Point]** Delete three items from the heap you created in (1) above. Show the deleted heap.
(3) **[1 Point]** Show the **max heap** that is built using **Bottom-Up** Algorithm on the data 11, 21, 31, 6, 46, 71, 81.

Your answer

1- 10,20,30,5,45,70,80

10,20,30,5,45,70,80

20,10,30,5,45,70,90

30,10,20,5,45,70,80

30,45,20,5,10,70,80

45,30,20,5,10,70,80

45,30,70,5,10,20,80

70,30,45,5,10,20,80

70,30,80,5,10,20,45

Final heap: 80,30,70,5,10,20,45

2 - 30,10,20,5

3- 1, 21, 31, 6, 46, 71, 81

1, 21, 81, 6, 46, 71, 31

81,21,1,6,46,71,31

81,21,71,6,46,1,31

Final heap: 81,46,71,6,21,1,31

2 points for (1) and (2)

81 will not reach the root.in line 3.

Q6 2 out of 3

Question 7. [3 Points]

(1) [**1 Point**] Write the fastest non-recursive algorithm you can think of, to find the **third** largest item in an array. You can assume array has n elements where n is greater than 2.

(2) [**1 Point**] What is its time complexity?

(3) [**1 Point**] What is the **reason** to your claim that this is the fastest non-recursive algorithm to solve this problem?

Your answer

```
if( arr_size<3)
{
System.out.printf("invalid input");
return;
}
// Find first

//largest element

int first = arr[0];

for ( int i = 1; i<arr_size; i++)

if(arr[i] > first)
first = arr[i];
// Find second

// largest element
int second = Integer.MIN_VALUE;
for(int i = 0; i<arr_size; i++)
if(arr[i] > second && arr[i] < first)
second = arr[i];

//Find third
// largest element
int third = Integer.MIN_Value;
for(int i = 0; i<arr_size; i++)
if (arr[i] > third && arr[i] < second)

System.out.printf(" The third largest element " + " element is %d\n")
```

2- The time complexity is O(n)

3- This is fastest because we have to at least see each element, which means it can't be faster than O(n).

Your answer uses three loops. It can be done using a single loop. At the very least (1) is wrong.

Q7 3 out of 3

[redacted]

Have a great life!