

DATA STRUCTURES

BATCH – A

[TUESDAY FEBRUARY 28, 2017: 2:00 PM – 5:00 PM]

ASSIGNMENTS – 7

CODE: assign07

INSTRUCTIONS:

[Total Marks: 20]

- i) Read all assignments and each problem has to be answered in the same c file.
- ii) Create a .c file following the file name convention: `abc-assign07.c`
Where `abc` is your roll number and `assign07` is the assignment code
- iii) Strictly follow the file name convention and do not use `scanf()`

PROBLEMS:

1) **[Marks: 4 marks]**

Define a node - `BTNODE` - of a binary tree with the following fields:

```
proID: <int> - [1000, 9999]
rank: <int> - [1, 9000]
level: <int> - it can range from [0, h] in every level where h is the
              height of the binary tree
cost: <float> - [99.0, 499.00]
discount: <float> - [2.0, 5.0]
```

The values of these fields could be generated using a random number generator in the specified range.

2) **[Marks: 16 marks]**

Using above data structure and the function prototypes given below, write your code for following tasks:

a) **[Marks: 4 marks]**

Create a binary tree with n nodes.

```
BTNODE *genBinaryTree(BTNODE *bnode, int n);
```

This function should internally insert an element into the binary tree in such a way that the resulting binary tree is complete.

b) **[Marks: 2 marks]**

Write a function to print `proID`, `rank` and price of each item:

```
void printElements(BTNODE *bnode);
```

c) **[Marks: 3 marks]**

Write a function to search and print the item by cost in a specified range
[129.0, 399.95]

```
void RangeSearchByCost(BTNODE *item, float scost);
```

d) **[Marks: 3 marks]**

Write a function to search an item with the lowest cost. Use this function to find and print the details of two nodes with the lowest cost.

```
void SearchMinCost(BTNODE *btnode);
```

e) **[Marks: 4 marks]**

Write a function to delete all elements at a specified level.

```
BTNODE *deleteElements(BTNODE*btnode, int level);
```