BRONX COMMUNITY COLLEGE

of The City University of New York

DEPARTMENT OF MATHEMATICS and COMPUTER SCIENCE

CSI 33

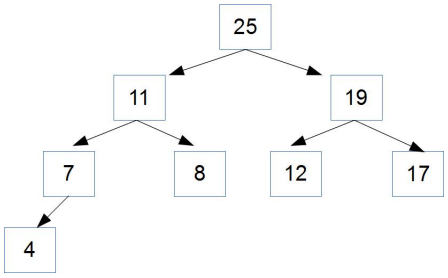
Final Exam Sample/outline

Programs must be sent to my e-mail : **natna20@gmail.com**. All other things are at your discretion: you can also send them via e-mail, or submit them in class. Note that everything must be submitted before you leave the class or the time is up.

1 Part 1

Do all problems in this part.

1. (10 points) Answer True/False and Multiple Choice questions (5 questions)
2. (15 points) Given the following heap:



   (a) Draw the array-based representation of this binary tree

   (b) Show addition of value 10 to the heap (and corrected heap)

   (c) Show deletion of value 25 from the heap (and corrected heap)

   references: Lecture 24 or 13.2 in the book

3. (5 points) The integers $20, 12, 7, 14, 2, 5, 3$ and 8 are inserted in that order into a priority queue. Give the order in which these values are retrieved.

   references: Lecture 23 or see Section 13.2, pages 444—445 in the book

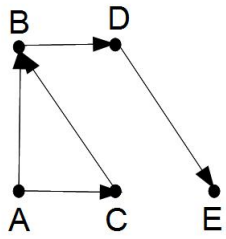
4. (5 points) The same integers (from the previous problem) are inserted into a binary search tree, one by one (initially the binary search tree was empty). Draw the final tree as it appears after all the insertions, without re-balancing after each insertion operation.

   references: Lecture 13 or see Section 7.5 in the book

5. (10 points) Do the same thing as in the previous problem for an empty AVL tree. After each insertion operation the tree needs to be re-balanced. Show all the work (as the tree appears after every insertion and after every re-balancing).

   references: Lecture 14 or see Section 13.3 in the book

6. (15 points) Represent this directed, unweighted graph as a Python list two ways: as an adjacency matrix and as an adjacency list. Which representation is most efficient for a graph which has very few edges between nodes?

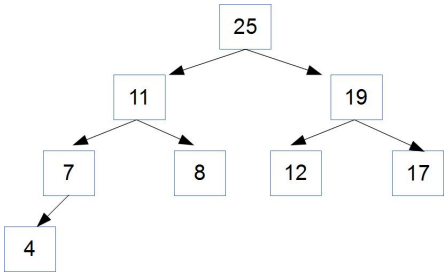


   references: Lecture 25 or see Section 14.2 in the book

2 Part II

Do any 2 problems.

1. (10 points) Given a binary tree, write the output of preorder, inorder, and postorder traversal.



   references: Lecture 13, HW#13: programming exercise 3, page 248 and Section 7.5.3, pages 238-239 in the book. The earlier encounter of it as *prefix*, *infix*

and *postfix* notations can be found in Section 5.2.4 of the book and Lecture 8 slides.

2. (10 points) Give box-and-arrow diagrams that illustrate the state of the C++ memory model imediately after each of these statements, including the last. What problem occurs in this program? How is it called? How can it be fixed?

```
int *b, *c;
b = new int;
*b = 3;
c = new int;
*c = 5;
*b += *c;
c = b;
delete c;
delete b;
```

references: Lecture 20 or see Sections 10.1 and 10.2 in the book

3. (10 points) The following C++ function finds the product of all integers in an array:

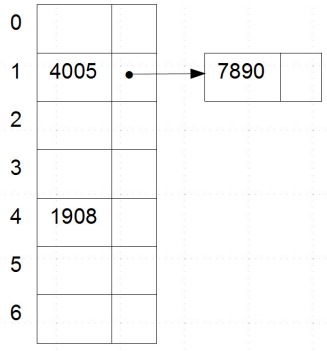
```
int product(const int *a, int size){

int i, prod=1;

for(i=0; i<size; i++){
   prod *= a[i];
}
return prod;
```

Rewrite this function as a template function which could be called as `product<int>`, `product<double>`, or as an instance of any other class which overloads the * operator.

references: Lecture 23 or see Section 12.2 in the book

4. (10 points) Below is an array with 7 positions, which is used as a hash table to keep some ids. The key to each record is the last 4 digits of customer's SSN. The hash function f is given by $f(k) = k\%7$, which gives the index of the slot in the array for the key k. The method of collision resolution is separate chaining. Draw the boxes and arrows in the following diagram to give the state of the hash table after the following keys are inserted in the order: 4005, 1908, 7890, 1928, 0035, 1076, 0187, 1098, 7777, 1108, 0089, 1625.

   As you can see the first three insertions have already been made as a hint.



   references: Lecture 24 or see Section 13.5 in the book

3 Part III

(20 points) You will be given an code of unfinished program and will be asked to finish it by adding or modifing attributes/methods