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. You are to answer True/False and Multiple Choice questions (10-11 questions)

1. Python lists are implemented using contiguous arrays. True or False?
2. Inserting into the middle of array-based implementation of list is a $\Theta(n)$ operation. True or False?
3. Looking up an item in a Python dictionary, given a key, is a $\Theta(n)$ operation. True or False?
4. Which of the following is **not true** of Python list (choose only one)?
   (a) They are implemented underneath as contiguous arrays
   (b) They allow for efficient random access
   (c) They can grow and shrink dynamically
   (d) All items in a list must be of the same type
5. What operation is not supported for Python dictionaries?
   (a) Item ordering (sorting)
   (b) Item insertion
   (c) Item deletion
   (d) Item lookup
6. When using a linked implementation of a queue, where should insertions be done?

(a) at the front (head) of the linked list

(b) in the middle of the list

(c) at the end (tail) of the list

(d) anywhere in the list, just keep the information of its location

7. A C++ function must return a value. True of False?

8. The *scope* of a variable refers to

   (a) the different values it can hold

   (b) the section of code where the variable can be accessed

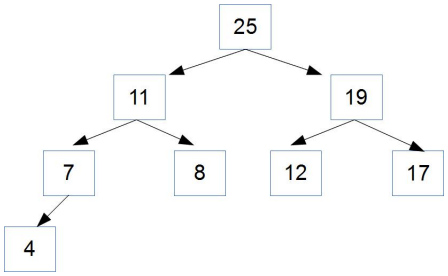
   (c) the time during which memory is allocated for the variable

   (d) the name of the variable.

2 Part 2

Answer all questions in this part (4-5 questions).

1. Given the following heap:



   (a) Draw the underlying array representation of this heap

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

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

   references: Lecture 24 or 13.2 in the book

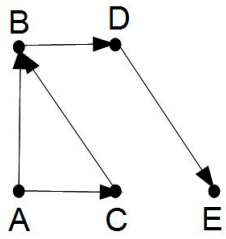
2. The integers $20, 12, 7, 14, 2, 5, 3$ and 8 are inserted in that order into a priority queue. Give the order in which these values are retrieved.

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

3. The same integers: $20, 12, 7, 14, 2, 5, 3$ and 8 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.

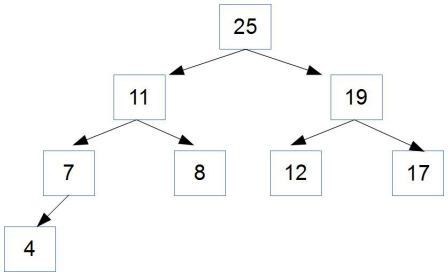
   references: Lecture 13 or see Section 7.5 in the book

4. 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

5. 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.

3 Part 3

Do any 3 problems out of 5.

1. The integers $20, 12, 7, 14, 2, 5, 3$ and 8 are inserted into an empty (initially) AVL tree, one by one. After some of the insertion operations 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

2. Give box-and-arrow diagrams that illustrate the state of the C++ memory model immediately 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

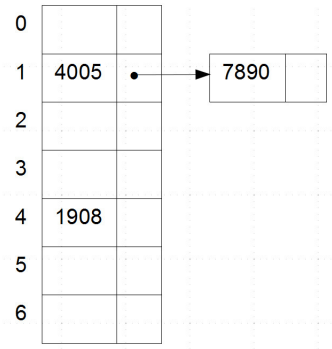
3. The following C++ function finds the product of all integers in an array:

```
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. 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

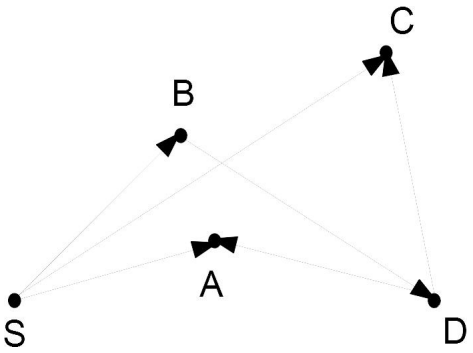


hash table after the following keys are inserted in the order: 4005, 1908, 7890, 1928, 0035, 1076, 0187, 1098, 7777, 1108, 0089, 1625.

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

5. For the given unweighted directed graph, use the unweighted shortest path algorithm if Breadth First Search. Assume that S is the source vertex.



Show all the work, including the queue and the table the algorithm produces. When finished, give the path from vertex S to vertex D.