# CST 370 Design and Analysis of Algorithms
# Spring 2020
# Final Exam

Name: ________________________________________

Four-digits ID: __________________________________

"On my honor, I have neither given nor received unauthorized aid in doing this assignment."

**Signature** [**Type Your Name**] ______________________________

- Do not start until told to do so.
- Look over all the questions and observe their point values before you start.
- Use your time wisely—make sure to answer the questions you know first.
- **Read the questions carefully.**

1. (2 points) (a) Assume that Dr. Byun assigned a programming project which requires the time complexity of $O(n^2)$. If your program's basic operation runs (2*n*logn + 25) times, can you say that your program meets the project requirement? (Yes/No).

(b) Consider the following algorithm.

```
1.  Algorithm Mystery(n)
2.  // Input: A nonnegative integer n
3.  S ← 0
4.  for i ← 1 to n do
5.      k ← i * i
6.      S ← S + k
7.  return S
```

Present the time complexity of the algorithm using the Θ notation. If you can't represent it using the Θ notation, indicate it clearly.

2. (1 point) Let $T(n) = 1/2*n^2 + 3*n$. Which of the following statements are true? (**Choose all that apply**.)

(a) $T(n) = O(n)$
(b) $T(n) = \Omega(n)$
(c) $T(n) = \Theta(n^2)$
(d) $T(n) = O(n^3)$

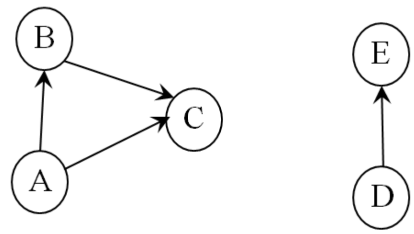
3. (3 points) Assume that **C(n)** indicates the **number of comparison** at the line number 3 of the following pseudocode. Using the notation C(n), present the **recurrence relation** and **initial condition** of the **number of comparison(s)**.

```
Algorithm Riddle(A[0..n − 1])
//Input: An array A[0..n − 1] of real numbers
if n = 1 return A[0]
else temp ← Riddle(A[0..n − 2])
     if temp ≤ A[n − 1] return temp
     else return A[n − 1]
```

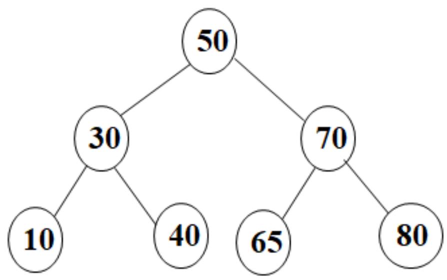
Initial condition: ___________________________

Recurrence Relation: ___________________________

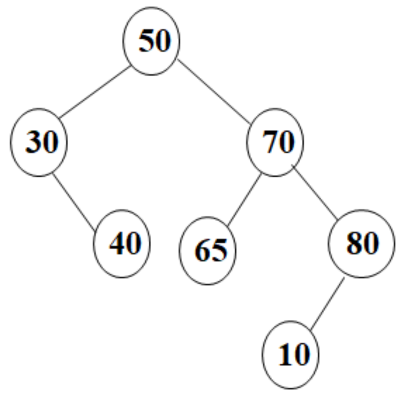
4. (2 points) (a) Is the following graph a DAG (= directed acyclic graph)? (Yes/No)




(b) Is this a 2-3 tree? (Yes/ No)



**[Note] Before solving the problem 5, read the following description carefully.**
In the problem 5, you have to present the result trees in the **level-by-level order**. This is an example of level-by-level order for a sample tree below. Note that the root value 50 is the level 0. Then, its children (= 30 and 70) should be the level 1. Also, because there's no value in the level 4 and 5, we use "NONE" to indicate them.




**A Sample Tree**

| | |
|---|---|
| Level 0 | **50** |
| Level 1 | **30, 70** |
| Level 2 | **40, 65, 80** |
| Level 3 | **10** |
| Level 4 | **NONE** |
| Level 5 | **NONE** |

**Level-By-Level Order**

5. (3 points) Consider a binary tree with ten nodes with the values 0, 1, …, 9 in such a way that the **inorder** and **preorder** traversals of the tree yield the following lists:

9, 3, 1, 4, 0, 2, 8, 6, 5, 7 (inorder)
2, 1, 9, 3, 0, 4, 8, 5, 6, 7 (preorder)

Note that the problem is asking to **consider only one binary tree**. For the problem, **do not draw the result in the word file**. Instead, **write the values of the result tree level-by-level order**.
If you think that it's not possible to have a binary tree with the given information, explain why.

| | |
|---|---|
| Level 0 | |
| Level 1 | |
| Level 2 | |
| Level 3 | |
| Level 4 | |

6. (5 points) Assume that you have five different data structures like below. In each data structure, there are *n* integer numbers and want to check if a specific number exists or not. In other words, you want to search a number in the data structure.
Write the **worst case time complexity** of the search operation in each data structure using the O notation.

| | |
|---|---|
| Unsorted array | |
| Sorted singly linked list | |
| Binary search tree | |
| AVL tree | |
| Hashing (Separate Chaining) | |

7. (5 points) Assume that you conduct the **linear probing** with the hash function ***h(K) = K mod 5***. This is the initial hash table for the problem. Note that the status '**E**' indicates "**Empty**".

| Index | Content | Status |
|---|---|---|
| 0 | | E |
| 1 | | E |
| 2 | | E |
| 3 | | E |
| 4 | | E |

Assume that you conduct the following six operations. Note that the **load factor** of the hashing for this problem is **0.5**.

1) **insert 2**
2) **insert 12**
3) **insert 22**
4) **insert 25**
5) **insert 30**
6) **delete 12**

(a) Present the hash table size after finishing the six operations: ______________

(b) This is a part of the hash table after finishing the six operations. Fill out it. For the status, use "E" for "Empty", "A" for "Active", and "D" for "Deleted".

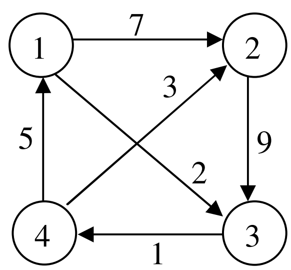
| Index | Content | Status |
|---|---|---|
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |

8. (2 points) (a) Assume that you should give 48 cents to a customer. How would you give the changes with the **least number of coins**? Choose the correct algorithm technique. ________

(1) Brute force algorithm
(2) Divide and conquer algorithm
(3) Dynamic programming algorithm
(4) Greedy algorithm

(b) Greedy algorithms always provide optimal solutions to their problems. (true / false)

9. (4 points) Apply Floyd's algorithm to find all-pairs-shortest-paths of the digraph defined by the following graph. For your understanding, $D^{(0)}$ is already provided. Fill out $\mathbf{D^{(1)}}$ and $\mathbf{D^{(2)}}$. Note that you **don't need to present and $\mathbf{D^{(3)}}$ and $\mathbf{D^{(4)}}$** for the problem.




$D^{(0)} =$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 7 | 2 | $\infty$ |
| 2 | $\infty$ | 0 | 9 | $\infty$ |
| 3 | $\infty$ | $\infty$ | 0 | 1 |
| 4 | 5 | 3 | $\infty$ | 0 |

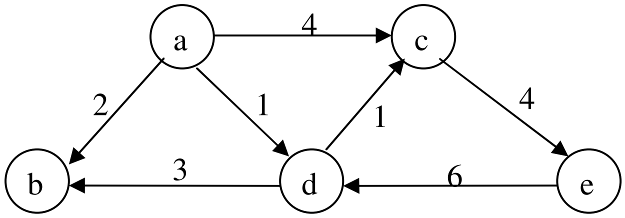
$D^{(1)} =$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

$D^{(2)} =$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

10. (5 points) Assume that you are going to solve the single-source shortest-paths problem using the **Dijkstra's algorithm** for the following graph. Note that you will start from the vertex **a**. For the problem, it's good enough for you to **present first three steps (= three rows of the table)**.




| **Vertex Visited** | **Remaining Vertexes** |
|---|---|
| | |
| | |
| | |

11. [**Puzzle**] (2 points) Assume that there are **two missionaries** and **two cannibals** in a river. They have to cross the river using a boat which can accommodate up to two people.

One constraint in the problem is that missionaries cannot be outnumbered by cannibals. For example, the situation of one missionary and two cannibals is unacceptable. But one missionary and one cannibal are fine.

Present the **minimum number of crossings of the boat** so that all four can cross the river.

For the problem, you **don’t need to explain your idea**. **Just write the minimum number of crossings**.

If they can't cross the river by the boat, write it clearly.