CST 370 Design and Analysis of Algorithms
SP'20 (Final Exam)

Name: _____Adam Ayala_____

Four-digits ID: _____2020_____

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

Signature [**Type Your Name**] _____Adam Ayala_____

- 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).

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
```

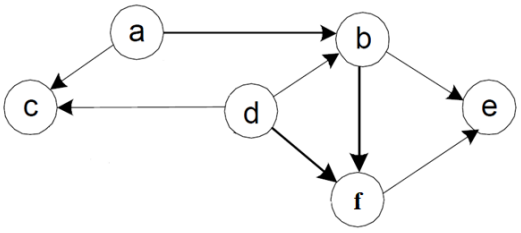
Present the time complexity of the algorithm using the Θ notation. If you can't represent it using the Θ notation, indicate it clearly.

O(n)

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

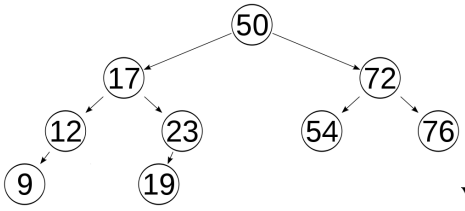
(a) $T(n) = O(n)$. **TRUE**
(b) $T(n) = \Omega(n)$. **TRUE**
(c) $T(n) = \Theta(n^2)$. **FALSE**
(d) $T(n) = O(n^3)$. **FALSE**

3. (2 points) (a) Is the following graph a DAG (= directed acyclic graph)? (Yes/No)



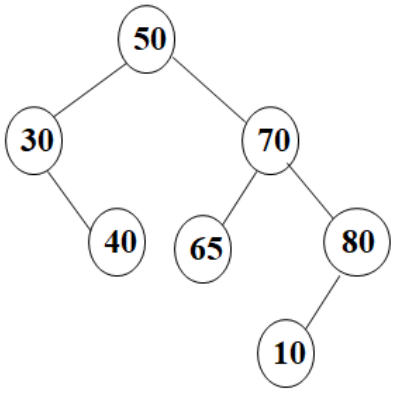
YES

(b) Is this an AVL tree? (Yes/ No)



YES

[Note] Before solving the problem 4, read the following description carefully.
In the problem 4, 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

4. (3 points) Consider a binary tree with ten nodes with the values 0, 1, …, 9 in such a way that the **inorder** and **postorder** traversals of the tree yield the following lists:

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

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	**2**
Level 1	**3, 8**
Level 2	**9, 0, 7, 5**
Level 3	**1, 4, 6**
Level 4	**NONE**

5. (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.

Sorted array	**O(log n)**
Unsorted singly linked list	**O(n)**
Binary search tree	**O(n)**
AVL tree	**O(log n)**
Hashing (Linear Probing)	**O(n)**

6. (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**.

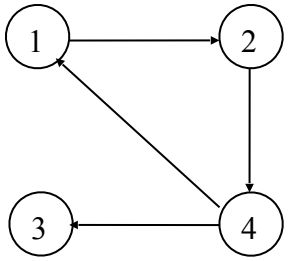
1) **insert 2**
2) **insert 12**
3) **insert 22**
4) **insert 30**
5) **delete 12**
6) **insert 25**

(a) Present the hash table size after finishing the six operations: _____11_____

(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	22	A
1	12	D
2	2	A
3	25	A

7. (4 points) Assume that you use the **Warshall's algorithm** to find the transitive closure of the following graph. For your understanding, $R^{(0)}$ is already provided. **Present $R^{(1)}$ and $R^{(2)}$.** Note that you **don't need to present and $R^{(3)}$ and $R^{(4)}$** for the problem.



$R^{(0)}$ =

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	0	0	0
4	1	0	1	0

$R^{(1)}$ =

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	0	0	0
4	1	1	1	0

$R^{(2)} =$

	1	2	3	4
1	0	1	0	1
2	0	0	0	1
3	0	0	0	0
4	1	1	1	1

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. _____

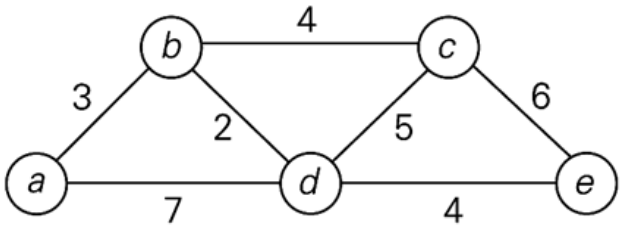
(1) Brute force algorithm
(2) Divide and conquer algorithm
(3) Dynamic programming algorithm
(4) Greedy algorithm

Greedy algorithm

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

FALSE

9. (5 points) Assume that you are going to solve the **minimum spanning tree (MST)** using the **Prim's algorithm**. Note that you start from the vertex **c** for the problem, and it is good enough for you to **present first three steps (= three rows of the table)**. Again, you should pay attention that **the starting vertex is "c", not "a"**.

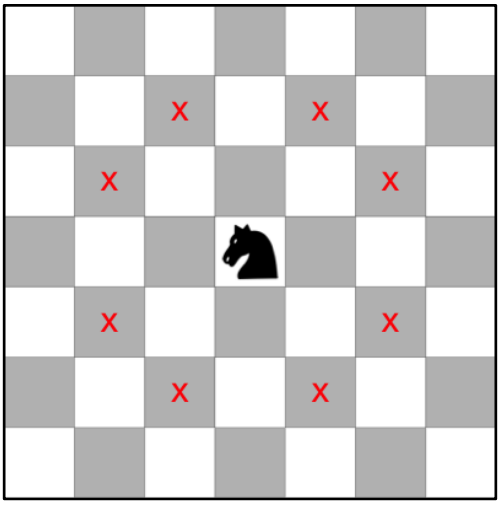


Vertex Visited	Remaining Vertexes
c(-, -)	a(-,∞) b(c,4) d(c,5) e(c,6)
b(c,4)	a(b,3) d(b,2) e(c,6)
d(b,2)	a(b,3) e(d,4)

10. [**Puzzle**] (2 points) What is the minimum number of moves needed for **a chess knight** to go from one corner of a 100 × 100 board to the diagonally opposite corner?

Note that the knight's moves are L-shaped: It can move two squares horizontally and one square vertically, or two squares vertically and one square horizontally in a single move. For example, let's assume that a chess knight is in the middle square of the following board. Then, it can move to 8 different squares as the diagram indicates.

For the problem, you **don't need to present your idea**. **Just write the minimum number of moves.**



66 moves

11. (3 points) Consider the following algorithm.

```
// Assume that n is a positive integer (= i.e. n >= 1), and A[1..n] is a global array.
// Note that the index of array A starts from one, not zero.
// And also, don't forget the array A in the algorithm is global.

Algorithm DoSomething (n)
1.  if (n = 1)
2.      Print the current content of the whole array A in a single line;
3.      Move the cursor to the next line.
4.  else
5.      for i ← 1 to n do
6.          DoSomething(n – 1);    // Recursive call.
7.          if n is odd
8.              swap A[1] and A[n];
9.          else
10.             swap A[i] and A[n];
11. return;
```

(a) Present the execution result of the algorithm where an array **A** has "**5, 7**"and *n* is 2. Note that **the sequence of output results is important**. Thus, you have to describe your answer clearly.

57
75

(b) Present the execution result of the algorithm where an array **A** has "**3, 5, 7**"and *n* is 3. Note that **the sequence of output results is important**. Thus, you have to describe your answer clearly.

357
537
735
375
573
753