CST 370 – Summer A 2020
Homework 2
Due: 05/05/2020 (Tuesday) (11:55 PM)

Name:_____Andrew Bell_____

ID: _____1138_____

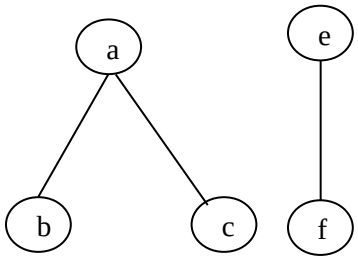
How to turn in?

Write your answer to the questions from 1 to 8 on a PDF file and submit it on the iLearn. Note that we **accept only a PDF** file. Do not submit a different file format. Also, don't forget to write your name and class ID at the top of your homework document.

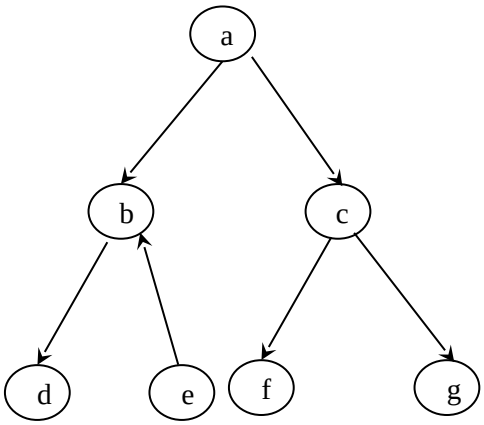
For the questions 9, 10, and 11, you should submit three source programs on the iLearn.

Thus, you have to submit total four files (one PDF file and three source files) on the iLearn.

1. (a) Based on our textbook's definition, is this a graph? (**True**/False) True



(b) Based on our textbook's definition, is this a graph? (**True**/False) True



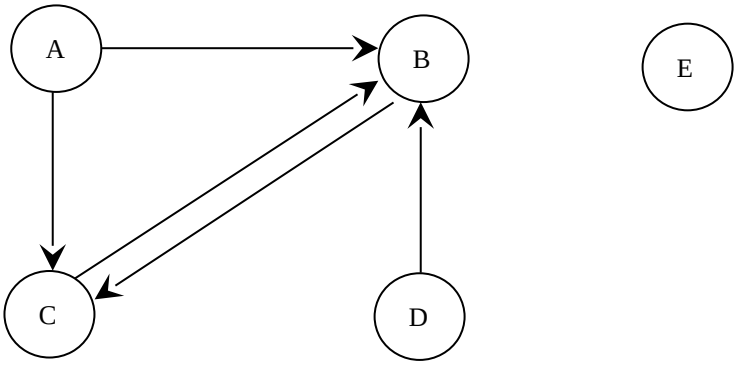
2. Assume that you should search a number in a list of *n* numbers. How can you take advantage of the fact that **the list is known to be sorted**? Give separate answers for the following two cases.

(a) A list represented in an array.
You can use the Binary search algorithm, checking if what you are looking for is bigger or smaller than the middle removing the other half (ignoring), and continuing to do this until you find the item.

(b) A list represented in a linked list.
Linked list doesn't have any ability for random access so the only optimization you could do is check if its closer to one end of the list.

3. Represent the following graph in the adjacency list as you learned in the class. Note that there are **five vertices** (= A, B, C, D, and E) in the graph.

[**Note**: If it is difficult to draw a diagram when doing homework, you can draw it on paper by hand. Then take a picture of the paper and insert it.]



A → B →C→ null
B → C→ null
C → B→ null
D → B→ null
E → null

4. Assume a binary tree with five vertices such as v1, v2, v3, v4, and v5. Determine the maximum number of edges possible in the tree.

4 edges per vertex * 5 vertices = 20 edges

5. (a) If your program takes n^2 time and your classmate's program takes $n*log\ n$ time, whose program is faster? Pick one between "You" and "Your Classmate".

Your Classmate, n*log n is faster.

(b) If your program takes *constant* time and your classmate's program takes *log n* time, whose program is faster? Pick one between "You" and "Your Classmate".

You, constant is faster

6. Consider the following algorithm.

```
Algorithm DoSomething (A[0.. n – 1])
1. num1 ← A[0];
2. num2 ← A[0]
3. i ← 1
4. while i < n do
5.     if A[i] < num1
6.         num1 ← A[i];
7.     if A[i] > num2
8.         num2 ← A[i];
9.     i ← i + 1
10. return (num2 – num1);
```

(a) Present the basic operation of the algorithm. When you present the basic operation, you should indicate the line number of the basic operation clearly.
The "<" on line 4.

(b) Present the time complexity category of the algorithm among the eight most popular time complexity categories we covered in the lecture.
O(n)

7. 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
```

(a) What does this algorithm compute?
Sum of all of the numbers squared up to n

(b) Present the time complexity category of the algorithm among the eight most popular time complexity categories we covered in the lecture.
O(n)

8. 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) $\mathbf{T(n) = \Theta(n^2)}$
(d) $T(n) = O(n^3)$

C

==========

This is the HackerRank link: https://www.hackerrank.com/cst370-su20-hw2

9. Write a program called **hw2_1.cpp (or hw2_1.java)** that reads a positive integer number from a user and displays the reverse of the number. For the program, you can assume that the input number is in the range of the typical "int" data type.

Sample Run 0: Assume that the user typed the following number.

```
1234321
```

This is the correct output.

```
1234321
```

Sample Run 1: Assume that the user typed the following number.

```
425
```

This is the correct output.

```
524
```

Sample Run 2: Assume that the user typed the following number.

```
1200
```

This is the correct output. Note that the reverse of 1200 is not 0021. It should be 21 because we should remove the leading zeros.

```
21
```

10. Write a program called **hw2_2.cpp (or hw2_2.java)** that reads two timestamps of two events from a user and displays the difference between the two timestamps. For the program, you can assume that each timestamp is composed of the hour (0 ~ 23), minute (0 ~ 59), and second (0 ~ 59) format. Your program should present the difference from the second event (= second timestamp) to the first event (= first timestamp). Note that **the second event always happens after the first event** and your program should display the time difference of the events.

Sample Run 0: Assume that the user typed the following two lines.

```
18:45:30
20:50:59
```

This is the correct output of your program.

```
02:05:29
```

Sample Run 1: Assume that the user typed the following two lines.

```
20:18:59
04:25:17
```

This is the correct output of your program.

```
08:06:18
```

Sample Run 2: Assume that the user typed the following two lines.

```
02:00:25
15:30:00
```

This is the correct output of your program.

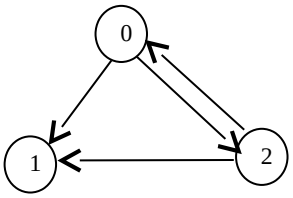
```
13:29:35
```

11. Write a program called **hw2_3.cpp (or hw2_3.java)** that converts a directed graph data from a user into a corresponding adjacency list format. Your program should read an input graph data from a user. Then, your program should convert it to the adjacency matrix format. In the assignment, you can **assume that the maximum number of vertices in the input graph is less than or equal to 50**.

Input format: This is a sample input from a user.

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

The first line (= 3 in the example) indicates that there are three vertices in the graph. The second line (= 4 in the example) presents the number of edges in the graph. The remaining four lines are the edge information in the graph. For the homework, you should assume that the first vertex starts from the number 0. Thus, **t1.txt** describes a directed graph like below:



Sample Run 0: Assume that the user typed the following lines

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

This is the correct output of your program.

```
0->1->2
1
2->0->1
```

Note that **the sequence of output values is important**. For example, when you display the neighbor vertices of the vertex 0, the sequence should be 1 and then 2 (= ascending order). If your program displays 2 and then 1, it's not correct. Similarly, for the vertex 2, your program should display 0 and then 1.

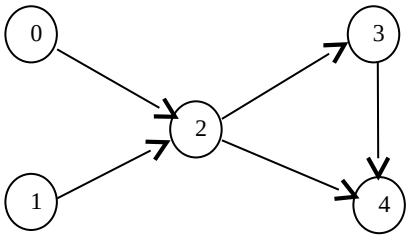
Sample Run 1: Assume that the user typed the following lines

```
5
5
2 3
3 4
1 2
0 2
2 4
```

This is the correct output of your program.

```
0->2
1->2
2->3->4
3->4
4
```

Note that this is the directed graph of the input data:



Sample Run 2: Assume that the user typed the following lines

```
3
6
0 1
1 0
1 2
2 1
2 0
0 2
```

This is the correct output of your program.

```
0->1->2
1->0->2
2->0->1
```