CST 370 – Spring A 2020
Homework 1

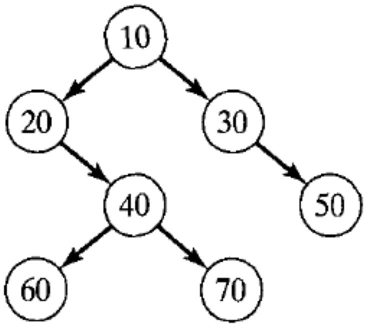
Name: ________________________________

Class ID: ______________________________

How to turn in?

- Write your answer to the question 1, 2, 3, and 4 and submit it on the iLearn. You can submit the file in PDF format. Don't forget to write your name and class ID number at the beginning of the file.
- For Question #5, you should submit your C++ source file (**hw1_1.cpp**) on the iLearn.
- Thus, you have to submit two files (one PDF file and one C++ source file) on the iLearn.
- Note that the due date is 11:55(PM). This is the iLearn's timestamp, not your submission time. Since there could be a long delay between your computer and iLearn, you should **submit early**.

1. (10 points) Is the following tree a binary tree? If so, is it a binary search tree?



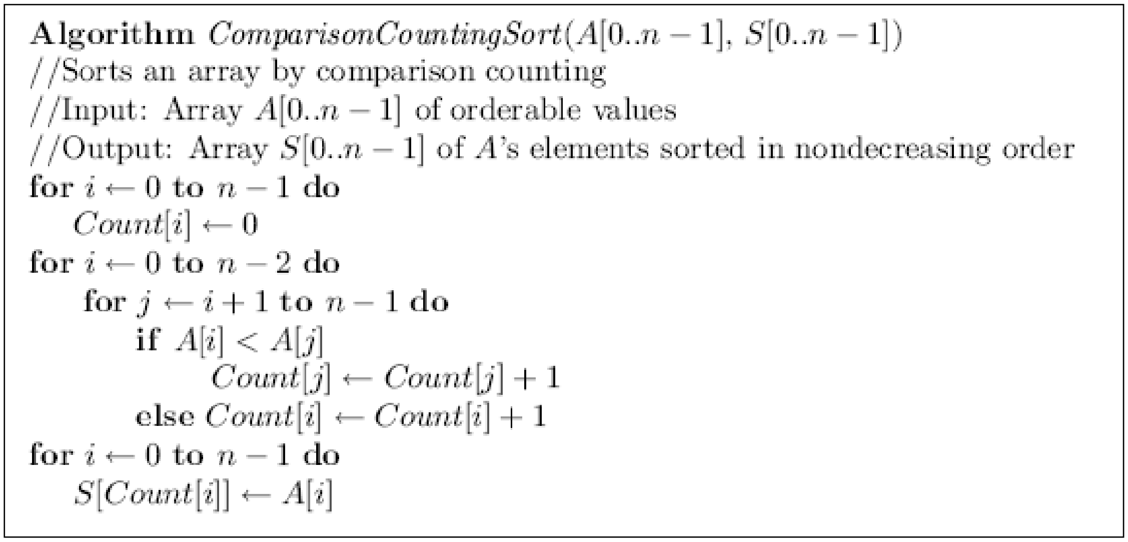
2. (10 points) (a) Find **gcd(270, 192)** by applying Euclid's algorithm as we covered in the class. Present the **intermediate steps** of the algorithm clearly.

(b) Find **gcd(270, 192)** by applying the middle school approach (= prime factors) as we covered in the class. Present the **intermediate steps** of the approach clearly.

3. (15 points) Consider the following pseudocode. **Describe what this algorithm computes.**

```
Algorithm Compute(A[0.. n – 1])
num1 ← A[0];
num2 ← A[0]
i ← 1
while i < n do
    if A[i] < num1
        num1 ← A[i];
    if A[i] > num2
        num2 ← A[i];
    i ← i + 1
return (num2 – num1);
```

4. (15 points) Consider the following algorithm to sort an array:



```
Algorithm ComparisonCountingSort(A[0..n − 1], S[0..n − 1])
//Sorts an array by comparison counting
//Input: Array A[0..n − 1] of orderable values
//Output: Array S[0..n − 1] of A's elements sorted in nondecreasing order
for i ← 0 to n − 1 do
    Count[i] ← 0
for i ← 0 to n − 2 do
    for j ← i + 1 to n − 1 do
        if A[i] < A[j]
            Count[j] ← Count[j] + 1
        else Count[i] ← Count[i] + 1
for i ← 0 to n − 1 do
    S[Count[i]] ← A[i]
```

(a) Apply the algorithm to sort the list 40, 50, 10. In other words, assume that the array **A[ ]** has the initial values 40, 50, and 10. Present the contents of the arrays **A[ ]**, **S[ ]**, and **Count [ ]** after finishing the execution of the algorithm.

(b) Similarly, present the contents of the three arrays if the array **A[ ]** has the initial values 60, 10, 60, 10, 20, 50?

5. (50 points) Write a C++ program called **hw1_1.cpp** that reads an input file name from a user. Note that the input file contains several integer numbers. After reading the numbers from the file, your program should display the closest distance between two numbers among the numbers. For example, if the input

numbers are -1, 10, 3, -15, 7, 20, 150, 200, 50, and 4, the answer should be 1 because the distance between 3 and 4 is 1. Note that the distance between two numbers is always positive.

Input file format: This is a sample input file called **`t1.txt`**.

```
5
2
1
2
-3
2
```

The first line (= 5 in the example) indicates that there are five integer numbers in the input file. The numbers from the second line (= 2 in the example) to the last line are actual numbers.

Sample Execution Result: For the assignment, your program has to **display the result exactly like the sample run**.

```
    Enter input file name: ./t1.txt

Output:

        Min distance: 1
        Two numbers for min distance: 1 and 2
```

Note that the input file name typed in bold such as **`./t1.txt`** is the user's input.

This is another sample input file called **`t2.txt`**.

```
8
-5
1
5
3
10
2
3
5
```

```
Enter input file name: ./t2.txt

Output:
        Min distance: 2
        Two numbers for min distance: 1 and 2
```

Note that You can get the sample t1.txt and t2.txt files on the iLearn.

For the program, you have to use a **dynamic memory** to create array(s) to store the input data. For details on the dynamic memory, read http://www.cplusplus.com/doc/tutorial/dynamic/

When you write the homework program, don’t forget to include five items such as “Title”, “Abstract”, “ID (= Four-digit number you selected for the class)”, “Name”, and “Date”. Refer to the following sample hello world program in C++.

```
/*
 * Title: hello_world.cpp
 * Abstract: This program displays a message Hello World on the screen.
 * Author: Dr. Byun [In your case, you should put your name.]
 * ID: 8899 [In your case, write your class-ID.]
 * Date: 04/12/2019 [In your case, put the date you write this program.]
 */

#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!\n";

    return 0;
}
```

Grading guide and test cases:

- Test case 1: Head comments (Title, Abstract, ID, Name, and Date)
  If you missed head comment elements, you will get penalty.

- Test case 2: Compilation.
  If the program failed to compile, it will be a serious issue.

- Test case 3:
  - t3.txt (input numbers)
    9
    2
    4
    6
    8
    10
    12
    14
    16
    1

  Output:
    Min distance: 1
    Two numbers for min distance: 1 and 2 (or 2 and 1)

- Test case 4:
  - t4.txt (input numbers)
    10
    -1
    -4
    -9

-16
-25
0
1
4
9
16

Output:

Min distance: 1
Two numbers for min distance: 0 and 1 (or 1 and 0)
Two numbers for min distance: -1 and 0 (or 0 and -1)

- Test case 5:

  -t5.txt (input numbers)

  10
  55
  99
  88
  77
  66
  50
  44
  33
  22
  11

  Output:

  Min distance: 5
  Two numbers for min distance: 50 and 55 (or 55 and 50)

- Test case 6:

  Dynamic array - We check if you use the dynamic array for the programming.