FINAL EXAMINATION

Course: **DATA STRUCTURES AND ALGORITHMS**

Term: **III** – Academic Year: **2020-2021** Class: **20CLC11**

Duration: **90** minutes Test date: **September 09, 2021**

Read this part carefully before solving the problems in this test

- You have to do your test by yourself. If you ask someone help to solve the problems or copy from the Internet, you will get **0** (for the whole course) because of cheating.

- You have to solve the problems by writing in your own prepared papers. Then you take the photos of your papers and combine all the photos to just **ONE (01) PDF file**.

- Each page must include your information (Student ID, Student Name), and page number at the top of the page.

- Name format of final test PDF file: **StudentID-DSA-Final.pdf**. Example: 20127999-DSA-Final.pdf.

PROBLEMS

Problem 01

Given the following integer array *a* with *n* elements ($0 \leq a_i \leq 100$) and the `sort(array, size)` function that can be used to sort an array ascendingly. Consider the following `doSomething()` function:

```
bool doSomething(int a[], int n){
     sort(a, n);
     for(int i = 0; i < n - 1; i++){
          if(a[i] == a[i + 1])
               return true;
     }
     return false;
}
```

a. *(5 pts)* Analyze the time complexity (using Big-O notation) of the `doSomething()` function.

b. *(15 pts)* What does the function `doSomething()` do? Design an algorithm with the time complexity of `O(n)` that does the same task as the `doSomething()` function but not changing the given array.

Problem 02

Design an algorithm (or propose an existing one) to sort the given array ascendingly that gives out the best performance (in time complexity) for each of the following array data orders:

a. *(10 pts)* An array *a* of size *n* with $a_i > a_{2i+1}$ and $a_i > a_{2i+2}$ (for every $i < n / 2$).

b. *(10 pts)* An real number array *a* already divided into an unsorted part of size *m* and a sorted part of size *n* ($m < n$).

Problem 03

Given an empty hash table with m = 17 slots. Hash function $h(k) = k \bmod m$ is used.

Show the result after inserting these values respectively: **16, 8, 90, 15, 12, 20, 28 4, 21, 22, 9, X** into the hash table. X is formed as follows: **1** and last 2 digits of your student ID. For example, if your student ID is 20127**01**, X will be **101**.

If there is a collision, use following collision resolution techniques:

a. *(10 pts)* Quadratic probing

b. *(10 pts)* Double hashing with h2(k) = (k mod 7) + 1

Problem 04

a. *(5 pts)* Given a binary search tree **T** which has 200 nodes with integer values in range [1, 500]. In case we need to search for value 45, which of the following choices is/are the sequence of compared nodes? Explain briefly.

A. 5, 2, 1, 10, 39, 34, 77, 63

B. 1, 2, 3, 4, 5, 6, 7, 8

C. 9, 8, 63, 0, 4, 3, 2, 1

D. 8, 7, 6, 5, 4, 3, 2, 1

E. 50, 25, 26, 27, 40, 44, 42

F. 50, 25, 26, 27, 40, 44

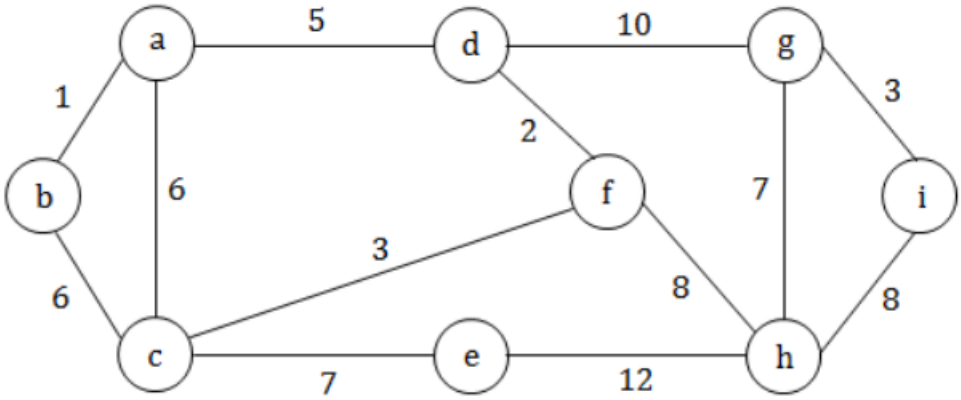
b. *(5 pts)* What order should we insert the following keys: **BACLIEU, CAMAU, SOCTRANG, CANTHO, HAUGIANG, VINHLONG, DONGTHAP, ANGIANG, KIENGIANG** into an empty AVL tree so that we do not have to perform any rotations on it? Show your resulting AVL tree.

c. *(5 pts)* Give the order of visited nodes when traversing your AVL tree (resulting from the above *question b*) pre-order and post-order.

d. *(5 pts)* How many nodes are there in an AVL tree with height of **10** having the minimum nodes? Explain your answer.

Problem 05

Given the following graph *G*



a. *(5 pts)* Give the order of visited vertices when traversing the above graph starting at **c** using depth-first search and breadth-first search strategies. Assumed that whenever we face with a decision of which node to pick from a set of nodes, pick the node whose label occurs earliest in the alphabet.

b. *(5 pts)* Use the Prim's algorithm to find the minimum spanning tree of graph *G*. What is the total cost of that tree?

c. (*10 pts*) Demonstrate the Dijkstra algorithm step by step to find the shortest paths (with cost) from node **i** to the remaining nodes of graph *G*.

THIS IS END OF THE TEST