Final DSA to students - Rbrbff

Algorithms and Data Structures (Trường Đại học Quốc tế, Đại học Quốc gia Thành phố Hồ Chí Minh)

International University - VNUHCM
School of Computer Science and Engineering

Final Examination

Date: 16/06/2021; Duration: 120 minutes

Online, Open-book

SUBJECT: Algorithms & Data Structures (IT013IU)	
Approval by The SCSE Signature  Full name: Dr. Nguyen Van Sinh	Lecturer: Signature  Full name: Trần Thanh Tùng
Proctor 1 Signature Full name:	Proctor 2 Signature Full name:
STUDENT INFO	
Student name: **Student ID:**	

INSTRUCTIONS: the total point is 100 (equivalent to 40% of the course)

1. *Purpose:*
   - Test your knowledge on data structures and algorithms in the following topics: Binary Tree, Hash Table, Graphs, Advanced graph algorithms
   - Examine your skill in analysis and design algorithms
2. *Requirement:*
   - Read carefully each question and answer it following the requirements
   - Write the answers and draw models CLEAN and TIDY directly in a **WORD** file
   - You can draw your trees, graphs by hand or by any tool (like draw.io)
   - Include the **SETTING** session below in your answer file.

Note: For all calculations in this subject, the following **rounding convention** is used: **7/2 = 4**

0. **Setting – TO INSERT TO YOUR ANSWER FILE**
   - **a.** Write the last 2 digits of your student ID (called is x):_____ (TO FILL IN)
   - **b.** Compute your **OFFSET** = x % 5 = _____ (TO FILL IN)
   - **c.** Using the table below to compute your **Starting node**: _____ (TO FILL IN)

OFFSET	**0**	**1**	**2**	**3**	**4**
STARTING NODE	**A**	**B**	**C**	**D**	**E**

Your list iteration procedure:

For all lists of items in the following sessions, take items from left to right starting from your **OFFSET.** If the end of the list is reached before all items are taken, continue from index 0 (wrap around).

1. **Binary search tree (25pts)**

Given a list of items, take items one by one using **your list iteration procedure**.

Table 1 - Items

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
45	30	80	20	35	70	90	25	21	39	37	38	75	78	76	90

1.a. Insert all items into a binary search tree and draw the tree (15pts)

1.b. Delete the root node and redraw the tree (10pts)

2. **Hash table (20pts)**

Given a list of items in table 1, take items using **your list iteration procedure**.

2.a. Insert all items into the hash table of size **27** by using the linear probing algorithm to solve collisions (10pts).

2.b. Change the hash table's size to **31**, redraw it (10pts)

3. **Graph - Elementary Algorithms (30pts)**

3.a Given the graph G_S in Fig. 1, run the DFS algorithm from your **STARTING NODE** and redraw the graph with the discovery time and the finishing time for each node. (15pts)

3.b. Find all strongly connected components in G_s and draw the G_s^{-1} with the finishing time for each node(15pts)

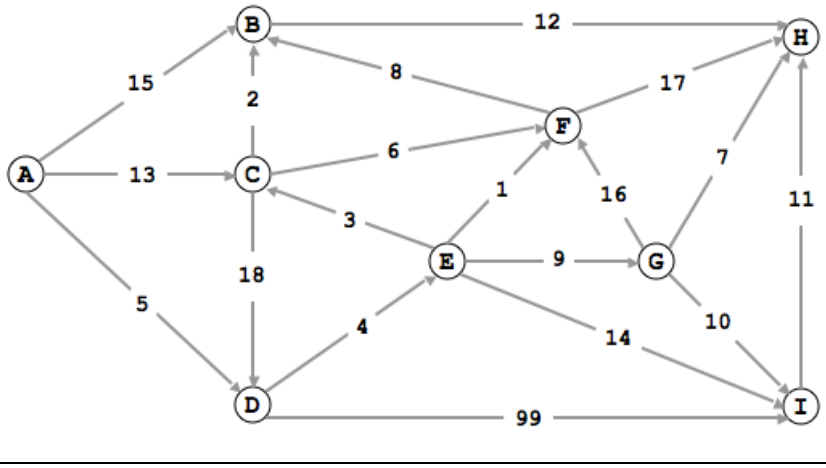


Fig. 1: Graph G_S

4. **Graph – Shortest path algorithm (15pts)**

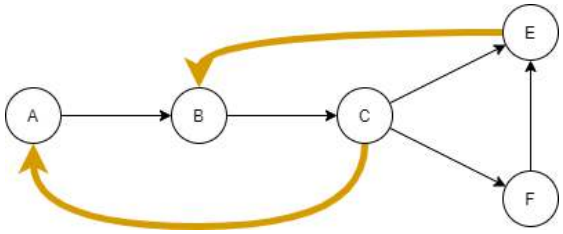
Run the Dijkstra's algorithm on the graph G_s in Fig.1 **from your starting node**, and fill the following table with corresponding values after each step of the algorithm

Selected nodes	A	B	C	D	E	F	G	H	I
						∞	∞	∞	∞

5. **Algorithm to find backward edges (10pts)**

While traversing a graph using the DFS algorithm, backward edges are edges that link a node to another node in the path from the source node to the node.

For example, in the graph below, colored edges are backward edges.



- (10pts) Propose an algorithm (write a pseudo-code) based on the DFS algorithm to print out all the backward edges of a given graph starting from a source node.

--- The end ---