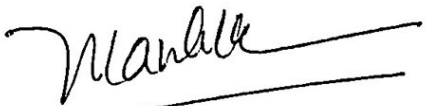



## FINAL EXAMINATION

16/06/2017

Duration: 90 minutes

Closed book –  
One A4 handwriting note is allowed

<b>SUBJECT:</b> <b>DATA STRUCTURE AND ALGORITHMS</b>	
School of Computer Science & Computer Engineering	Lecturer: Tung
Signature: 	Signature: 
Full Name: <u>Trần Mạnh Hà</u>	Full Name: Tran Thanh Tung

Student Name:

Student ID:

**Rounding convention:  $9/2 = 5$ . Nodes visiting convention: alphabetic order.**

1. Given a graph  $G$  represented by the following list of successors.

For each pair  $(x, y)$ ,  $x$  is the weight of an edge,  $y$  is a terminal extremity of the edge.

$A \rightarrow (5, B) \rightarrow (4, D) \rightarrow (1, E)$

$B \rightarrow (3, C) \rightarrow (1, F)$

$C \rightarrow (1, G)$

$D \rightarrow (9, C) \rightarrow (1, H)$

$E \rightarrow (5, H) \rightarrow (1, I)$

$F \rightarrow (1, J)$

$G \rightarrow (4, F) \rightarrow (1, K)$

$H \rightarrow (10, G) \rightarrow (1, L)$

$I \rightarrow (9, J)$

$J \rightarrow (3, K)$

$K \rightarrow (2, L)$

$L \rightarrow (6, I)$

- i. Draw the graph (15pts)
  - ii. Show the adjacency matrix of the graph (10pts)
2. Traverse the graph G using the BFS algorithm. Write down the list of nodes in the visited order (20pts).
3. Each edge in the graph G in question 1 is named in such a way that if there is an edge from X to Y, then the edge is named XY.
- i. List all edges of G in alphabetic order of its name and fill in the following table (5pts)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Edge																			
Weight																			

- ii. Use quick sort – median of 3 to sort all edges of G in ascending order with respect to its weight. Fill the following table with each step of the quick sort (20pts)

Action	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

4. In the graph G, use Dijkstra algorithm to compute shortest paths from A to all other nodes. Fill the following table with corresponding values after each step of the algorithm (20pts)

Node	A	B	C	D	E	F	G	H	I	J	K	L
d	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

5. Rooted trees with unbounded branching

Given a tree whose node may have arbitrary numbers of children. There is a schema to represent that kind of tree name *left-child, right-sibling*. Each node  $x$  contains a parent pointer  $p$ , and two other pointers:

- *left-child*[ $x$ ] points to the leftmost child of node  $x$ , and
- *right-sibling*[ $x$ ] points to the sibling of  $x$  immediately to the right.

If node  $x$  has no children, the *left-child*[ $x$ ] = NULL, and if node  $x$  is the rightmost child of its parent, then *right-sibling*[ $x$ ] = NULL.

- i. Give an example of a rooted tree with unbounded branching and draw its *left-child, right-sibling* representation (5pts).
- ii. Write an  $O(n)$  non-recursive procedure that prints all the keys of an arbitrary rooted tree with  $n$  nodes, where the tree is stored using the left-child, right-sibling representation (5pts)