Family Name: . . . .
Other Names: . . . .

Signature: . . . .
ID Number: . . . .

COMP 261 Test

18 April 2017

Instructions

- Time allowed: **45 minutes** .
- Answer **all** the questions. There are 45 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- This test contributes 20% of your final grade
- You may use paper translation dictionaries, and non-programmable calculators.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	**Marks**	
1. Graphs	[20]	□
2. Minimum Spanning Trees	[15]	□
3. Graphics	[10]	□
	TOTAL:	□

Question 1. Graphs [20 marks]

(a) [3 marks] For each of the following actions on a *simple graph*, identify the better data structure between *Adjacent Matrix* and *Adjacent List*.

- List all edges
- Find all outgoing neighbours of a node
- Find out whether there is an edge between two given nodes

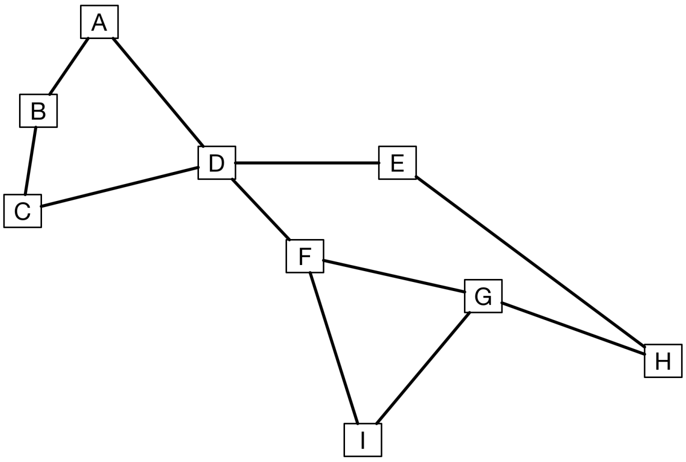
A *simple graph* means that there is at most one edge between any pair of nodes in the graph. In *Adjacency Matrix*, each element of the matrix is an edge object. In *Adjacency List*, each node is associated with a set of neighbouring nodes.

(b) [2 marks] Describe a situation that the graph cannot be represented by either Adjacency Matrix or Adjacency List.

(Question 1 continued on next page)

(Question 1 continued)

(c) [6 marks] In the graph below, the nodes are labeled as A, B, ..., I. The edges are all undirected. Give the *adjacency list* representation of the graph.

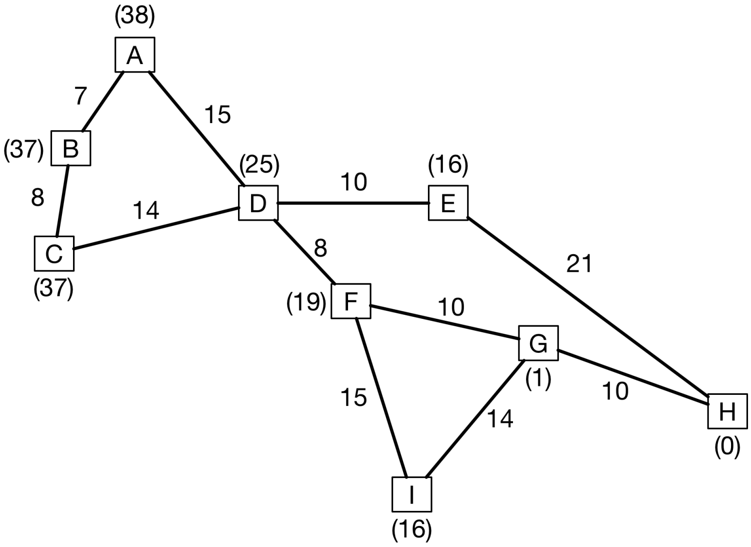


(Question 1 continued on next page)

(Question 1 continued)

(d) [9 marks] Given the graph below, the nodes are labeled as A, B, ..., I. Each edge is undirected, and is associated with a number indicating the edge length. Each node is associated with a number (in parenthesis) indicating the estimated distance to the node H. Show how to use A* search algorithm to search for the shortest path from node C to node H. You should show (1) at each step, the *elements in the fringe* and the *element to be visited next*, and (2) the *final shortest path* as a sequence of nodes.

Each element is represented in the format of $\langle node, fromNode, lengthSoFar, priority\rangle$. The information in step 0 is already given for you to start.



Step 0: Fringe elements: $\{\langle C, null, 0, 37\rangle\}$
Element to visit next: $\langle C, null, 0, 37\rangle$

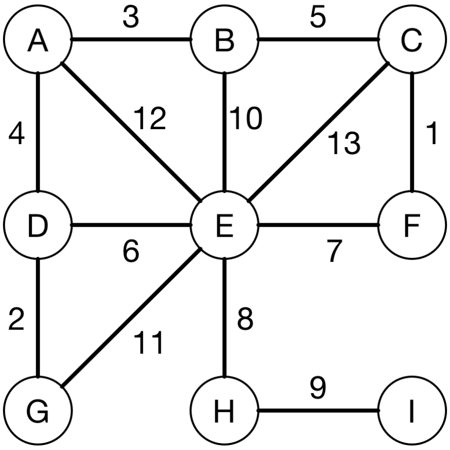
Final Path:

Question 2. Minimum Spanning Trees [15 marks]

(a) [5 marks] Given the following graph, show the steps Prim's algorithm would take to find a Minimum Spanning Tree (MST). The algorithm starts at node A. At each step, the algorithm adds one edge into the MST. List the edges in the order that they are added to the MST.

Each edge is represented as the two nodes that they connect and the edge weight: e.g. "EF 4" for the edge between E and F with weight 4.

To make the answer unique and easy to mark, list the nodes in each edge in *alphabetical* order (e.g. DE not ED).



Alphabet:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

(Question 2 continued on next page)

(Question 2 continued)

(b) [5 marks] Given the same graph, show the steps Kruskal's algorithm would take to find a MST. List the edges in the order that they are added to the MST.

Again, list the nodes in each edge in *alphabetical* order (e.g. DE not ED).

Alphabet:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

(Question 2 continued on next page)

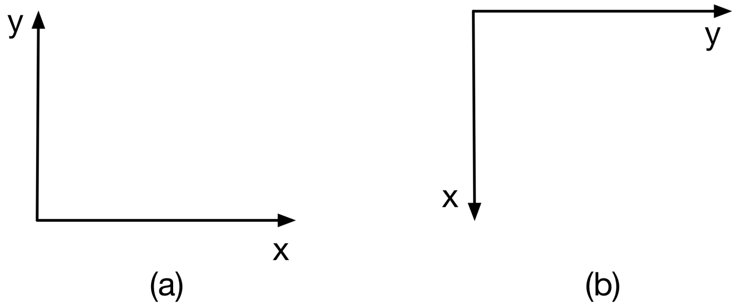
(Question 2 continued)

(c) [5 marks] The Union-Find data structure is very efficient in Kruskal's algorithm. It has three methods: MakeSet(x), Find(x) and Union(x,y). The pseudo code of MakeSet(x) and Find(x) methods are given below. Write the pseudo code of Union(x,y) that never merges longer tree into shorter tree.

```
MakeSet(x) {
    x.parent = x;
    x.depth = 0;
    Add x into the set;
}
Find(x) {
    if (x.parent == x)
        return x;
    else {
        root = Find(x.parent);
        return root;
    }
}
```

Question 3. Graphics [15 marks]

(a) [2 marks] When applying the right-hand rule to two graphs (a) and (b) given below, describe whether the z axis is pointing outward or inward, and give your reason.



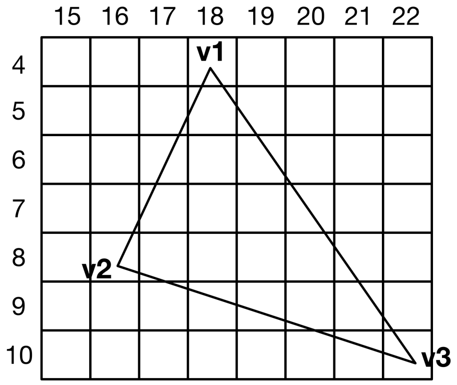
(Question 3 continued on next page)

(Question 3 continued)

(b) [8 marks] A triangular polygon in the x-y plane is given below. The x and y coordinates of the three vertices are given below:

- v1 = (18, 4),
- v2 = (16, 8),
- v3 = (22, 10),

show the output of the first step of the Edge List algorithm, that is, the Left and Right x values for each y value. Round each coordinate to 1 decimal place as appropriate.



x	
Left	Right

* * * * * * * * * * * * * * *

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.