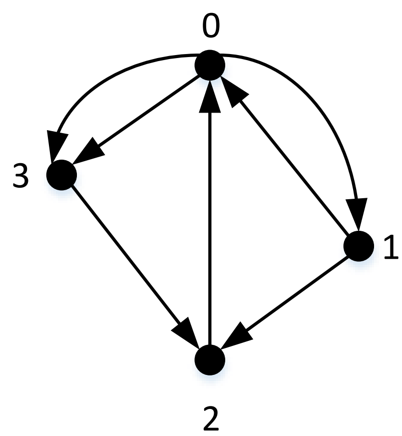
PART A – GRAPH THEORY – 20 MARKS

1. <u>Matrices in Graph Theory (10 marks)</u>

This question is based on the following graph G:



a) Fill out the adjacency matrix for the graph G:

↗	0	1	2	3
0	0	1	0	2
1	1	0	1	0
2	1	0	0	0
3	0	0	1	0

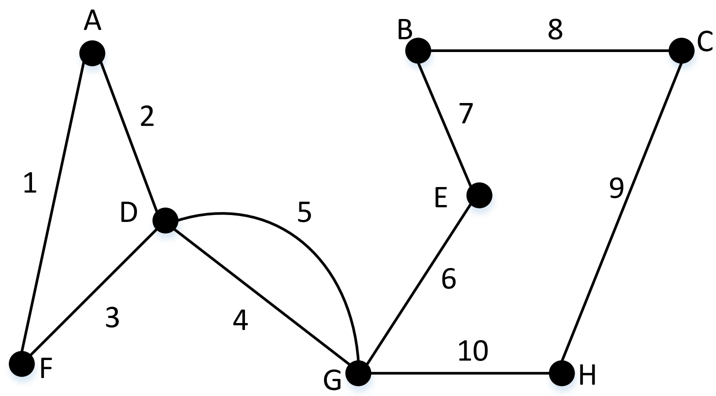
b) Fill out the following matrix A which is defined as follows:
A(i,j) = number of walks of length 2 from vertex i to vertex j in the graph G.

↗	0	1	2	3
0	1	0	3	0
1	1	1	0	2
2	0	1	0	2
3	1	0	0	0

This can be derived by squaring the adjacency matrix.

2. Circuits (6 marks)

This question is based on the following graph G (the edge numbers are edge names):



a) Give an Euler circuit for G (listing the vertices and edges as they are traversed) or explain why this cannot be done.

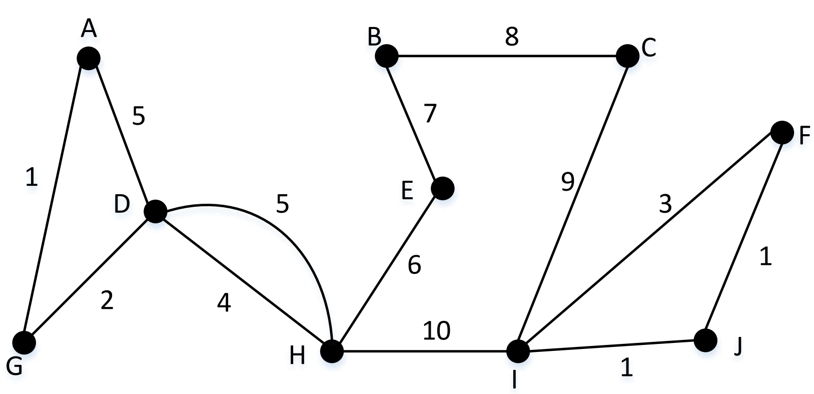
Starting at A, there are 8 Euler circuits as each of the cycles ADF,DG, GEBCH can be traversed either clockwise or counter clockwise. The one that is fully clockwise is: A2D5G6E7B8C9H10G4D3F1A

b) Give a Hamiltonian circuit for G (listing the vertices and edges as they are traversed) or explain why this cannot be done.

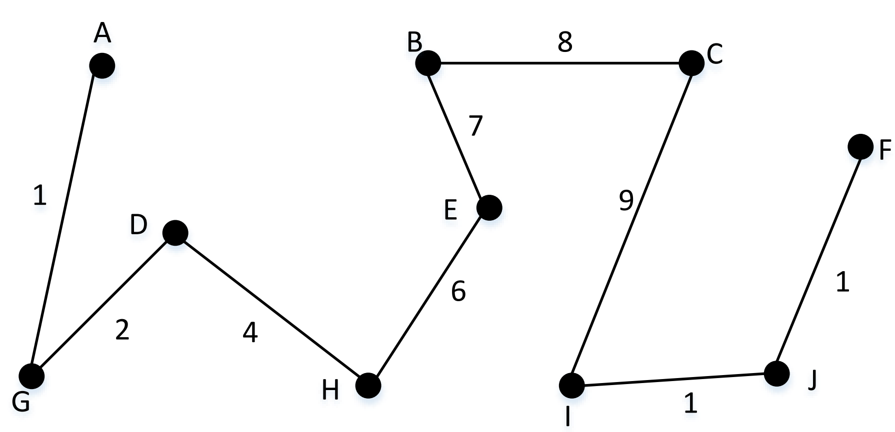
This graph consists of 3 simple circuits: ADF, DG, GEBCH with are connected at vertices D and G. **Any** circuit for the entire graph (i.e. including all the vertices) will need to traverse both D and G more than twice in order to include both the ADF and the GEBCH circuits. Therefore none of the circuits which include all the vertices will be Hanmiltonian circuits. .

3. Minimum Spanning Tree (4 marks)

For the weighted graph G underneath, where the edge numbers are weights:



Draw a minimum spanning tree (draw the edges you are keeping with their weights).



PART B – REGULAR EXPRESSIONS AND FINITE STATE AUTOMATA – 40 MARKS

1. Operations on Languages (10 marks)

Define the following two languages of the alphabet $\Sigma=\{a,b\}$:
$L_1 = \{a, aa, ab\}$
$L_2 = \{b, bb, ab\}$

a) List all the elements of $L_1 \cap L_2$

{ ab }

b) List all the elements of $L_1 \cup L_2$

{ a, aa, ab, b, bb }

c) List all the elements of $L_1 \times L_2$

{ (a,b), (a,bb), (a,ab), (aa,b), (aa,bb), (aa,ab), (ab,b), (ab,bb), (ab,ab) }

d) List all the elements of $L_1 L_2$

{ ab, abb, aab, aabb, aaab, abbb, abab }

2. Regular Expression (10 marks)

Write a regular expression to match all integers in a new programming language. Integers are defined as follows:

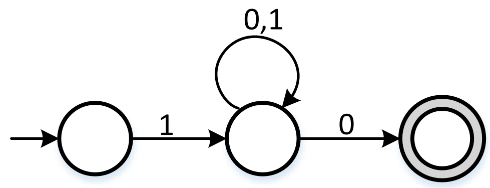
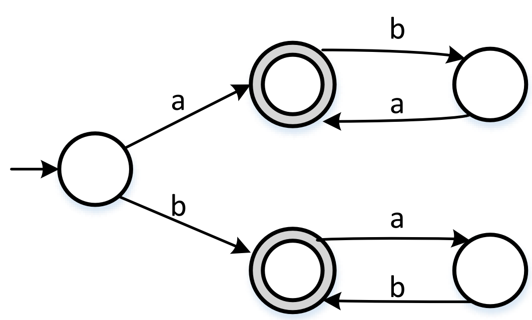
- An integer can be in base ten (decimal), base sixteen (hex) of base eight (octal).
- A decimal integer is either the single digit 0 or a sequence of one or more digits between 0 and 9 such that the leading digit is not a 0.
- A hex number starts with the string "0x" which is then followed by one or more digits between 0 and 9 or letters between A and F.
- An octal number starts with the single digit 0 which is then followed by one or more digits between 0 and 7.

You do **not** need to simplify your regular expression

(0 | [1-9][0-9]*) | (0x [0-9,A-F]+) | (0 [0-7]+)

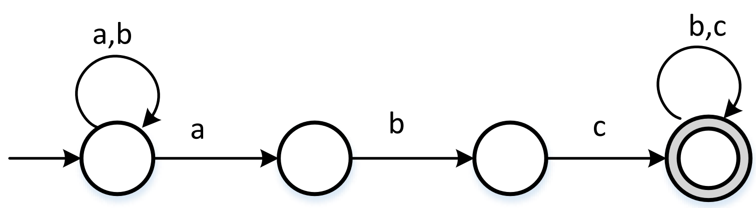
3. Finite State Automata (20 Marks)

a) Give a regular expression for each of the following finite state automata. Make these regular expressions as simple as possible.

Automaton	Regular expression
	1 (0|1)* 0
	(a (ba)*) | (b (ab)*)

In the next two questions the simplest possible automaton refers to an automaton with as few states as possible.

b) Draw the simplest possible NFA (non-deterministic finite state automaton) on an input alphabet I={a,b,c} which recognizes the following regular expression: (a|b)*abc(b|c)*



c) Draw the simplest possible DFA (deterministic finite state automaton) on an input alphabet I={a,b,c} which recognizes the following regular expression: (a|b)*abc(b|c)*