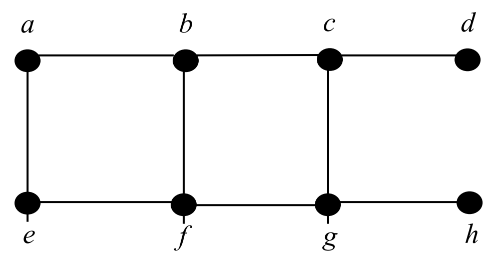
INT102 Algorithmic Foundations
Problem Session 2, Week 6
Group1: 09:00-11:00, 31/03/2023, Friday
Group2: 17:00-19:00, 31/03/2023, Friday
Location: SC176/Online

Suggested Solutions

Question 1

Consider the following graph G.



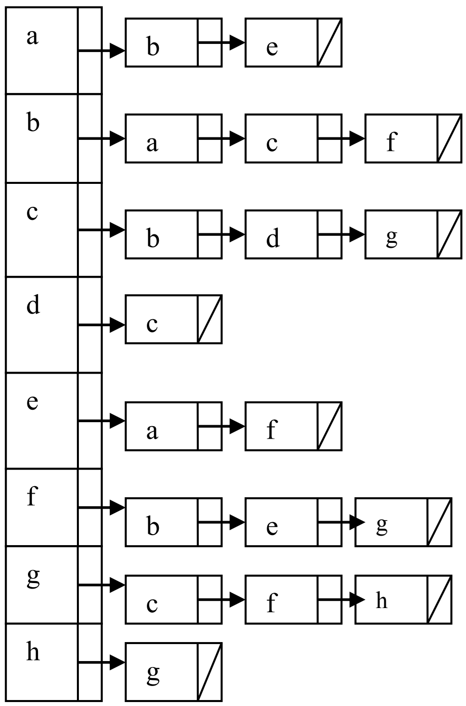
1. Give the adjacency matrix and adjacency list of the graph G. **(4 marks)**

Answer:

adjacency matrix (2 marks)

	a	b	c	d	e	f	g	h
a	0	1	0	0	1	0	0	0
b	1	0	1	0	0	1	0	0
c	0	1	0	1	0	0	1	0
d	0	0	1	0	0	0	0	0
e	1	0	0	0	0	1	0	0
f	0	1	0	0	1	0	1	0
g	0	0	1	0	0	1	0	1
h	0	0	0	0	0	0	1	0

adjacency list (2 marks)



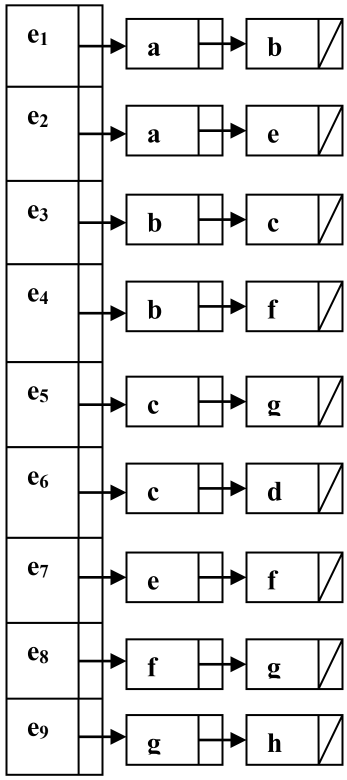
2. Give the incidence matrix and incidence list of the graph G. **(4 marks)**

Answer: Let $e_1=(a, b)$, $e_2= (a, e)$, $e_3= (b, c)$ $e_4= (b, f)$, $e_5= (c, g)$, $e_6= (c, d)$, $e_7= (e, f)$, $e_8= (f, g)$, $e_9= (g, h)$

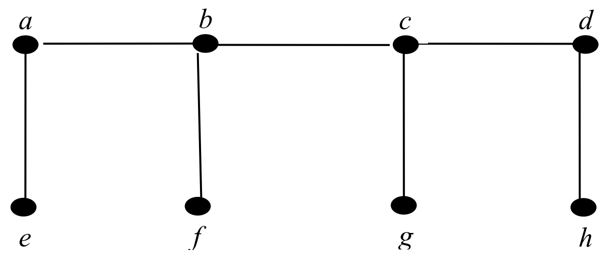
incidence matrix (2 marks)

	a	b	c	d	e	f	g	h
e_1	1	1	0	0	0	0	0	0
e_2	1	0	0	0	1	0	0	0
e_3	0	1	1	0	0	0	0	0
e_4	0	1	0	0	0	1	0	0
e_5	0	0	1	0	0	0	1	0
e_6	0	0	1	1	0	0	0	0
e_7	0	0	0	0	1	1	0	0
e_8	0	0	0	0	0	1	1	0
e_9	0	0	0	0	0	0	1	1

incidence list (2 marks)

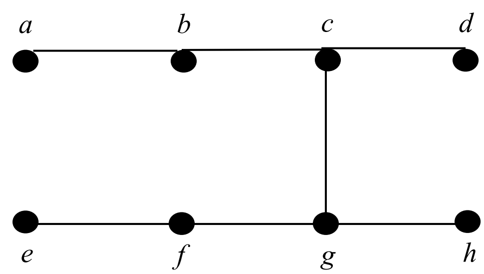


3. Starting at the vertex ***a*** and resolving ties by the vertex alphabetical order traverse the graph by breadth-first-search (BFS) and construct the corresponding BFS tree. **(5 marks)**



4. Starting at the vertex ***a*** and resolving ties by the vertex alphabetical order traverse the graph by depth-first-search (DFS) and construct the corresponding BFS tree. **(5 marks)**

Answer:



Question 2
Consider the following recursive function

$$f(n)=\begin{cases}1 & if \quad 0 \le n \le 3 \\ f(n-1)+f(n-2)+f(n-3)+f(n-4) & if \quad n>4\end{cases}$$

1. Write a recursive (top-down) algorithm to compute it. (**5 marks)**
   Answer:

```
Procedure F(n)
    if n==0 or n==1 or n =2 or n =3 then
            return 1
    else
            return F(n-1) + F(n-2) + F(n-3) + F(n-4)
```

2. What is the complexity of your algorithm (in big-O notation)? (**5 marks)**
   Answer:

$$\begin{aligned} f(n) &= f(n-1) + f(n-2) + f(n-3) + f(n-4) \\ &= [f(n-2)+f(n-3)+ f(n-4) + f(n-5)]+f(n-2)+ f(n-3) + f(n-4) \\ &> 2\, f(n-2) \\ &> 2\, [2\, f(n-2-2)] = 2^2\, f(n-4) \\ &> 2^2\, [2\, f(n-4-2)] = 2^3\, f(n-6) \\ &\ldots \\ &> 2^k\, f(n-2k) \end{aligned}$$

   If n is even, $f(n) > 2^{n/2} f(0) = 2^{n/2}$
   If n is odd, $f(n) > f(n-1) > 2^{(n-1)/2}$

   In any case, we have $f(n)=O(2^n)$

3. Design and write the pseudo code of a faster nonrecursive (bottom-up) algorithm using the concept of dynamic programming.**(5marks)**

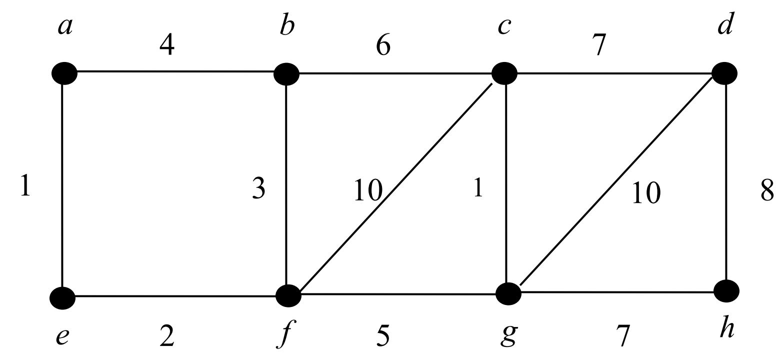
```
Procedure F(n)

    Set A[0] = A[1] = A[2] = A[3] =1

    for i = 4 to n do

            A[i] = A[i-1] + A[i-2] + A[i-3] + A[i-4]

    return A[n]
```

4. What is the time complexity of the faster algorithm (in big-O notation)? (**5 marks**)

   O(n)

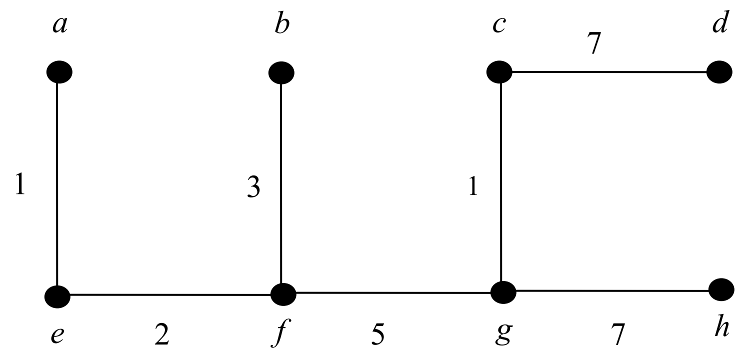
Question 3

Consider the following graph G. The label of an edge is the cost of the edge.



1. Using *Prim's* algorithm, draw a *minimum spanning tree* (MST) of the graph Also write down the change of the priority queue step by step and the order in which the vertices are selected. Is the MST drawn unique? (i.e., is it the one and only MST for the graph?) **[5 marks]**

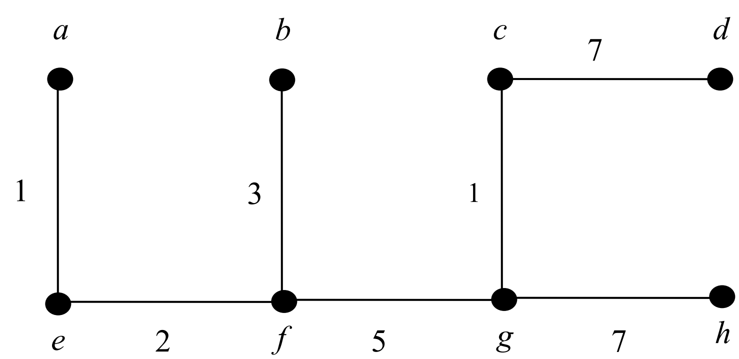
Order Selected	**a(0,-)**	**b(-,∞)**	**c(-,∞)**	**d(-,∞)**	**e(-,∞)**	**f(-,∞)**	**g(-,∞)**	**h(-,∞)**
a(0,-)		**b(a,4)**	**c(-,∞)**	**d(-,∞)**	**e(a,1)**	**f(-,∞)**	**g(-,∞)**	**h(-,∞)**
e(a,1)		**b(a,4)**	**c(-,∞)**	**d(-,∞)**		**f(-,2)**	**g(-,∞)**	**h(-,∞)**
f(e,2)		**b(f,3)**	**c(f,10)**	**d(-,∞)**			**g(f,5)**	**h(-,∞)**
b(f,3)			**c(b,6)**	**d(-,∞)**			**g(f,5)**	**h(-,∞)**
g(f,5)			**c(g,1)**	**d(g,10)**				**h(g,7)**
c(g,1)				**d(c,7)**				**h(g,7)**
d(c,7)								**h(g,7)**
h(g,7)								



This is the only MST.

2. Using *Kruskal's* algorithm, draw a *minimum spanning tree* (MST) of the graph G.
   Write down the order in which the edges are selected.
   Is the MST drawn unique? (i.e., is it the one and only MST for the graph?)
   (5 marks)

(a, e)	1/
(c, g)	1/
(e, f)	2/
(b, f)	3/
~~(a, b)~~	~~4~~
(f, g)	5/
~~(b, c)~~	~~6~~
(c, d)	7/
(g, h)	7/
~~(d, h)~~	~~8~~
~~(c, f)~~	~~10~~
~~(d, g)~~	~~10~~



This is the only MST.

3. Referring to the same graph above, find the shortest paths from the vertex ***a*** to *all* other vertices in the graph G using *Dijkstra's* algorithm. Show the changes of the priority queue step by step and give the order in which edges are selected. **(5 marks)**

N.B. There may be more than one solution. You only need to give one of the solutions.

Order Selected	**a(0,-)**	**b(-,∞)**	**c(-,∞)**	**d(-,∞)**	**e(-,∞)**	**f(-,∞)**	**g(-,∞)**	**h(-,∞)**
a(0,-)		**b(a,4)**	**c(-,∞)**	**d(-,∞)**	**e(a,1)**	**f(-,∞)**	**g(-,∞)**	**h(-,∞)**
e(a,1)		**b(a,4)**	**c(-,∞)**	**d(-,∞)**		**f(e,3)**	**g(-,∞)**	**h(-,∞)**
f(e,3)		**b(a,4)**	**c(f,13)**	**d(-,∞)**			**g(f,8)**	**h(-,∞)**
b(a,4)			**c(b,10)**	**d(-,∞)**			**g(f,8)**	**h(-,∞)**
g(f,8)			**c(g,9)**	**d(g,18)**				**h(g,15)**
c(g,9)				**d(c,16)**				**h(g,15)**
d(c,16)								**h(g,15)**
h(g,15)