CED19I027

I hereby pledge that I have not copied from anyone of my classmates/ seniors or any other sources. (Put tick mark if you are taking the pledge)

Due: 22nd Nov 2020; '-4' marks for late submission

---

1. Consider the following two statements.

I There exists a connected graph with 7 vertices and 5 edges. II There exists a tree with 8 vertices and 9 edges.

Which of the following is true.

(a) I and II are true (b) I is true and II is false

(c) I is false and II is true **(d) I and II are false**

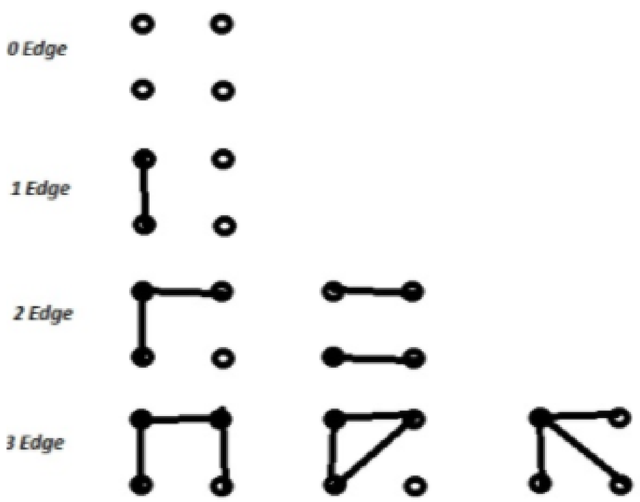
sol)
(d)
For a graph of n vertices to be connected , it should have a minimum of n-1 edges.
=> A graph with 7 vertices should have a minimum of 6 vertices to be connected .
=> Statement 1 is wrong
Statement 2 is wrong

---

2. Find the number of nonisomorphic graphs with 4 vertices and at most 3 edges.

(a) 6 **(b) 7** (c) 8 (d) 8

sol)
(b) 7



=> 7

---

3. Find the minimum and the maximum number of edges in any graph with 5 vertices and 2 components.

(a) 3 (b) 4 (c) 5 **(d) 6**

MIN**(a)**3 MAX**(d)**6

sol)

Min : (a) 3 Max : (d) 6

Given graph has 5 vertices and 2 components .

Let the number of vertices in component 1 be a_1
Let the number of vertices in component 2 be a_2
Now , $a_1 + a_2 = 5$

Minimum number of edges:
Now we look at every single completed component of the graph.
Any component with ‘n’ vertices has ‘n-1’ number of edges.
Therefore,

Minimum number of edges $= \sum_{i=1}^{2}(a_i - 1) = (a_1-1) + (a_2-1)$
$= a_1 + a_2 - 2$
$= 5 - 2 = 3$

∴ Minimum number of edges = 3

Maximum number of edges :
The maximum number of edges in a component with ‘n’ vertices is ${}^{n}C_2$
Therefore,

Maximum number of edges $= \Sigma_{i=1}^{2} ({}^{a_i}C_2) = \Sigma (a_i)(a_i-1)/2$
$= \Sigma [a_i^2 - a_i]/2$

Max. no of edges $= \sum_{i=1}^{2} \frac{(a_i^2 - a_i)}{2}$

$= \frac{a_1^2 + a_2^2 - a_1 - a_2}{2}$

$= \frac{(a_1+a_2)^2 - (a_1+a_2) - 2a_1a_2}{2}$

$= \frac{(a_1+a_2)[a_1+a_2-1] - 2a_1a_2}{2}$

$= \frac{5\times4 - 2a_1a_2}{2}$

$= 10 - a_1a_2$

~~For 10-a~~ For ‘$10 - a_1a_2$’ to be maximum, a_1a_2 should be minimum

$\Rightarrow a_1a_2 = 1\times4$
$= 4$

$\Rightarrow$ Max $= 10 - 4$
$= 6$

∴ Maximum no of edges = 6

4. Find the minimum and the maximum number of edges in any connected acyclic graph with 6 vertices

(a) 3 (b) 4 **(c) 5** (d) 6

MIN**(c)5** MAX**(c)5**

Sol)

(c) 5 for both minimum and maximum

A connected acyclic graph is nothing but a TREE.

A connected tree with 'n' vertices should have EXACTLY 'n-1' vertices because ,

if it has its number of edges > n-1 , it means there is a link between atleast 2 vertices in such a way that they become cyclic . so Number of edges cannot be greater than n-1

if it has its number of edges < n-1 , it means the graph is disconnected ,which shouldn't be the case.

So number of edges cannot be less than n-1

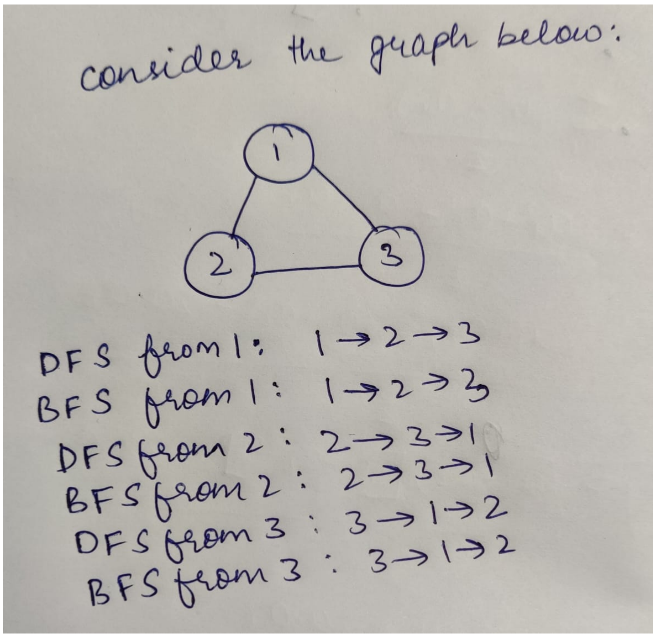
thus, both the maximum and minimum number of edges in any connected acyclic graph with n vertices = n-1

---

5. Give an example of a graph which gives same BFS and DFS for a particular starting vertex

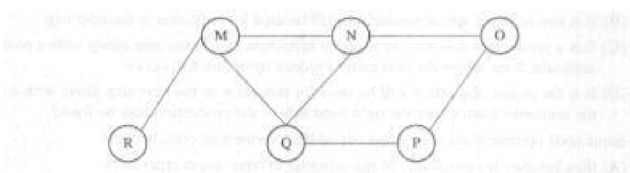
Sol)

Any graph with its number of vertices less than or equal to 3 have same BFS and DFS traversal



---

6. The BFS algorithm has been implemented. One possible order of visiting the nodes of the following graph is



(a) MNOPQR (b) NQMPOR **(c) QMNPRO** (d) QMNPOR

Sol)
Option (c) is one possible solution

---

7. Level order traversal of a rooted tree can be done by starting from the root and performing

(a) preorder traversal (b) inorder traversal (c) DFS **(d) BFS**

Sol)
Option (d) BFS
Level order traversal is nothing but traversal from the parent to all of its children,then to all of its grandchildren ,then to all of its great grandchildren and so on
BFS traversal is the traversal from one vertex , to all of its adjacent vertices , and so on.
Since both the traversals are similar , BFS can be used to perform Level order Traversal.

---

8. The weight of a minimum spanning tree of the following graph is **93**

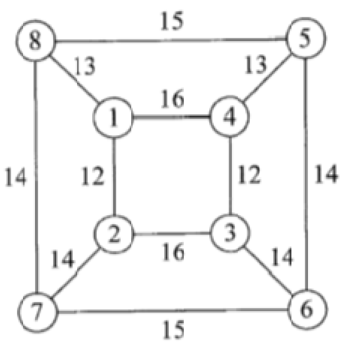
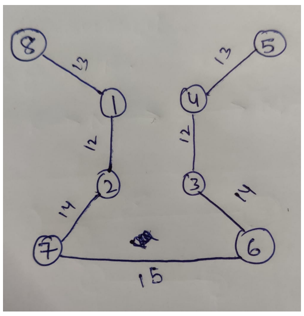


Fig. 4-2



12 + 12 + 13 + 13 + 14 + 14 + 15 = **93**

---

9. Suppose a circular queue of capacity (n-1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are

(A) Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT

(B) Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR

(C) Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT

(D) Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

Sol) Option (A)

---

10. Which one of the following array represents a binary max-heap? (1 Marks)

(A) {25, 12, 16, 13, 10, 8, 14}

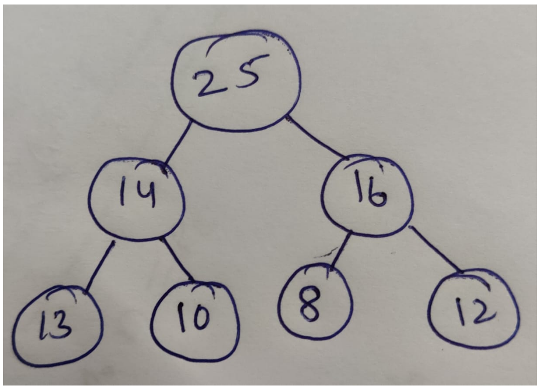
(B) {25, 14, 13, 16, 10, 8, 12}

(C) {25, 14, 16, 13, 10, 8, 12}

(D) {25, 14, 12, 13, 10, 8, 16}

Sol) ------------

option (c)



---

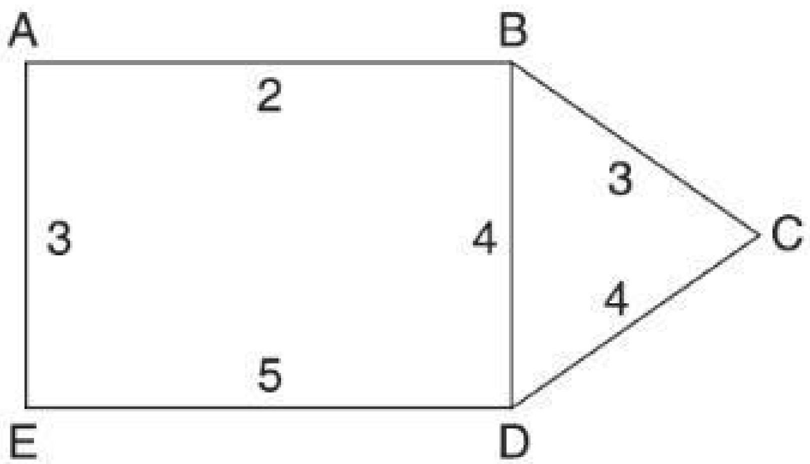
11. Where will be a minimum element found in a max heap?

(A) Within rst $\frac{n}{2}$ levels
(B) Within last $\frac{n}{2}$ levels
(C) leaves
(D) Level above leaves

ANS) Option (c) Leaves

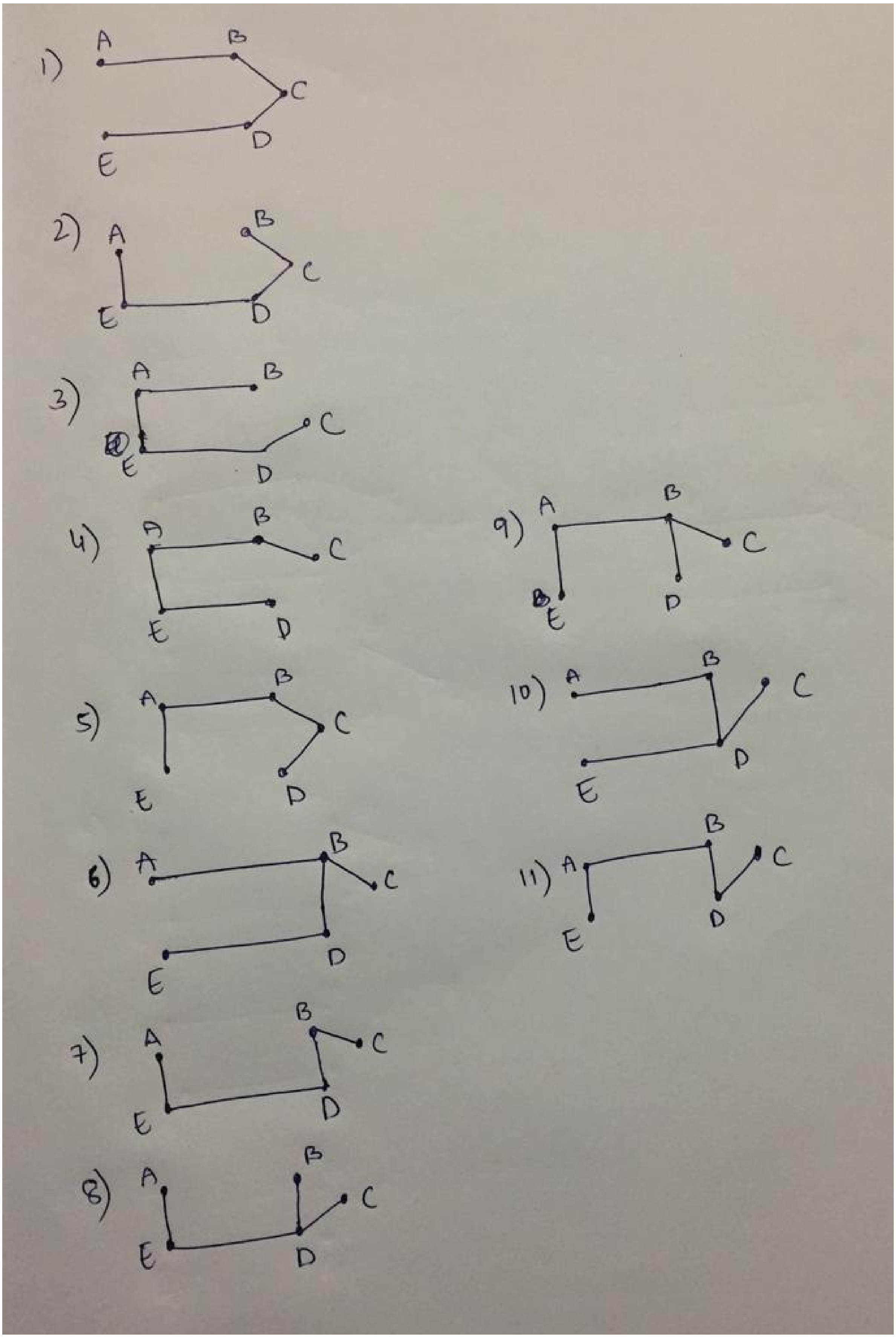
---

12. How many spanning trees does the following graph have? Draw them.



SOL)
The given graph has 11 possible spanning Trees and they are present as shown below:

Graphs- Hashing
Due on 27th Nov 2020



13. Consider an undirected unweighted graph G. Let a breadth- firrst traversal of G be done starting from a node r. Let d(r; u) and d(r; v) be the lengths of the shortest paths from r to u and v respectively in G. If u is visited before v during the breadth- rst traversal, which of the following statements is correct?

(a) d(r; u) < d(r; v)

(b) d(r; u) > d(r; v)

(c) d(r; u) <= d(r; v)

(d) None of the above

SOL)
For breadth first traversal ,
if u is visited before v , the possible cases are :
u and v must both be adjacent vertices to atleast one vertex [d(r;u) = d(r;v)]
or
u must be closer to r when compared to v [d(r;u) < d(r;v)]
Option (c) is the correct answer

---

14. Consider a weighted undirected graph with vertex set V = fn_1; n_2; n_3; n_4, n_5; n_6g and edge

set E = f(n_1; n_2; 2); (n_1; n_3; 8); (n_1; n_6; 3); (n_2; n_4; 4), (n_2; n_5; 12); (n_3; n_4; 7); (n_4; n_5; 9); (n_4; n_6; 4)g. The third value in each tuple represents the weight of the edge speci ed in the tuple.

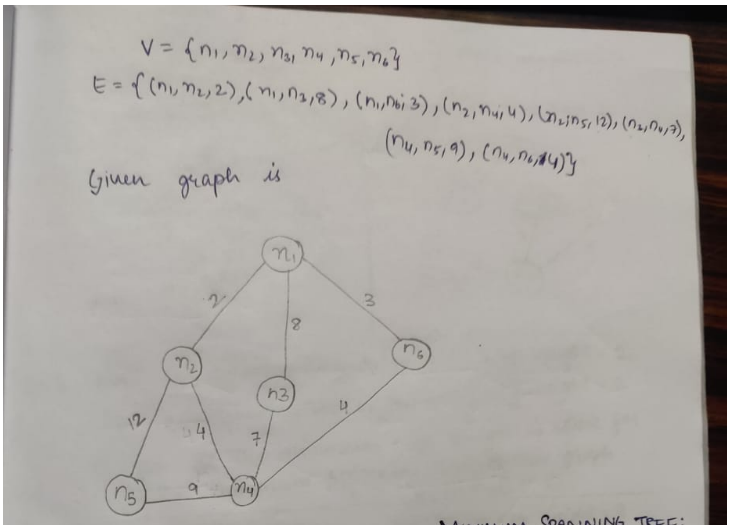
(A) List the edges of a minimum spanning tree of the graph.

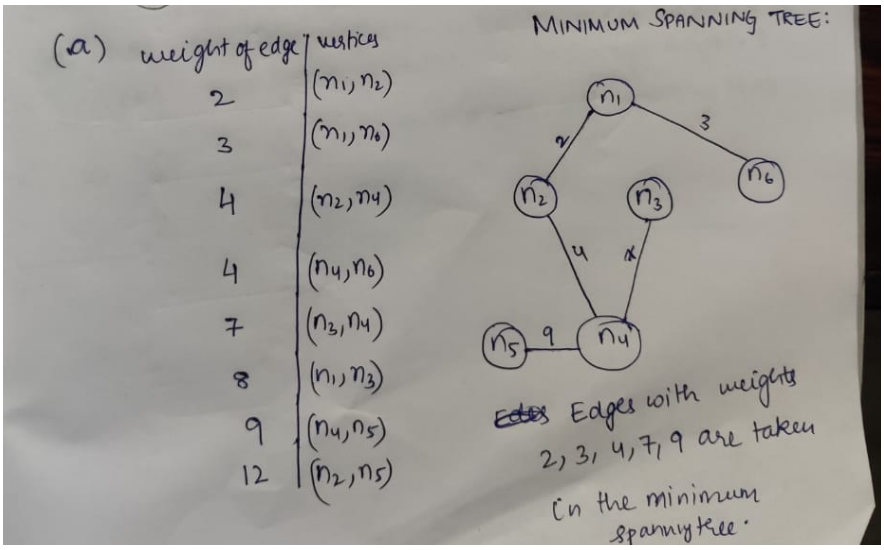
(B) How many distinct minimum spanning trees does this graph have?

(C) Is the minimum among the edge weights of a minimum spanning tree unique overall possible minimum spanning trees of a graph?

(D) Is the maximum among the edge weights of a minimum spanning tree unique over all possible minimum spanning trees of a graph?

Sol)



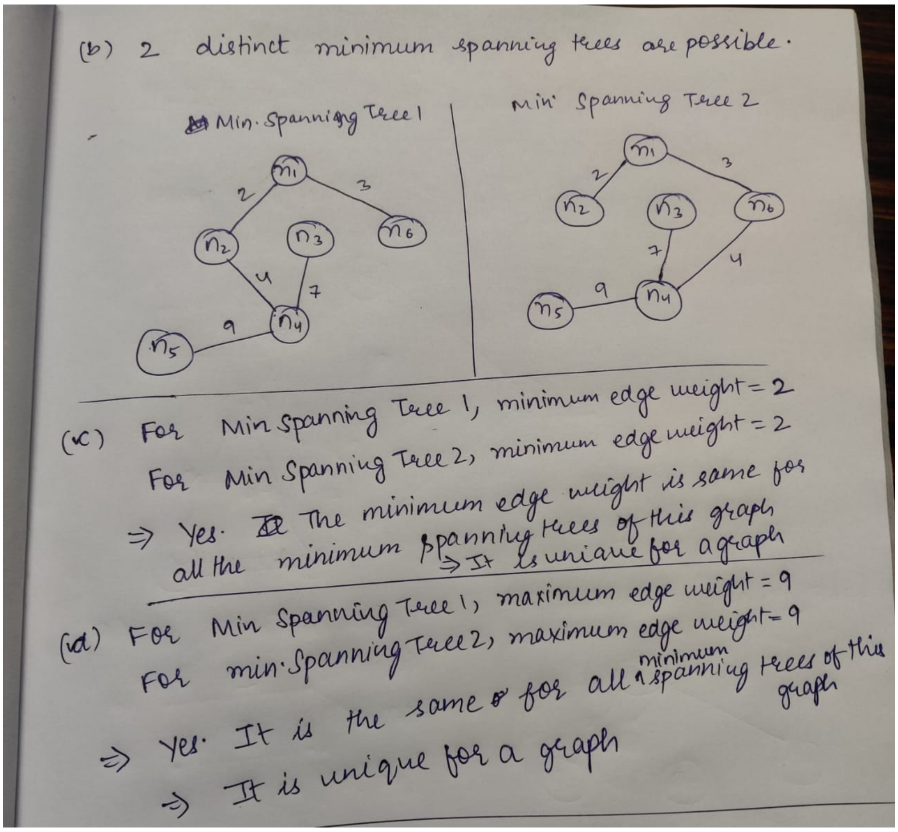


(a)

weight of edge	vertices
2	(n_1, n_2)
3	(n_1, n_6)
4	(n_2, n_4)
4	(n_4, n_6)
7	(n_3, n_4)
8	(n_1, n_3)
9	(n_4, n_5)
12	(n_2, n_5)

MINIMUM SPANNING TREE:

Edges with weights 2, 3, 4, 7, 9 are taken in the minimum spanning tree.



(b) 2 distinct minimum spanning trees are possible.

Min. Spanning Tree 1

Min Spanning Tree 2

(c) For Min Spanning Tree 1, minimum edge weight = 2
For Min Spanning Tree 2, minimum edge weight = 2
⇒ Yes. The minimum edge weight is same for all the minimum spanning trees of this graph
⇒ It is unique for a graph

(d) For Min Spanning Tree 1, maximum edge weight = 9
For min. Spanning Tree 2, maximum edge weight = 9
⇒ Yes. It is the same for all minimum spanning trees of this graph
⇒ It is unique for a graph

---

15. The minimum number of edges in a connected cyclic graph on n vertices is:

(A) n - 1

(B) n

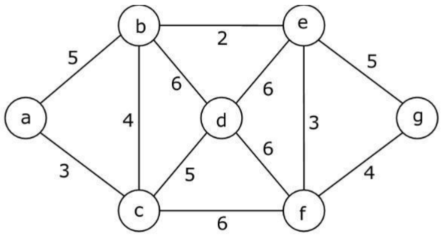
(C) n + 1

(D) None of the above

sol) option (B) n

---

16. Consider the following graph: Which one of the following is NOT the sequence of edges



added to the minimum spanning tree using Kruskal's algorithm?

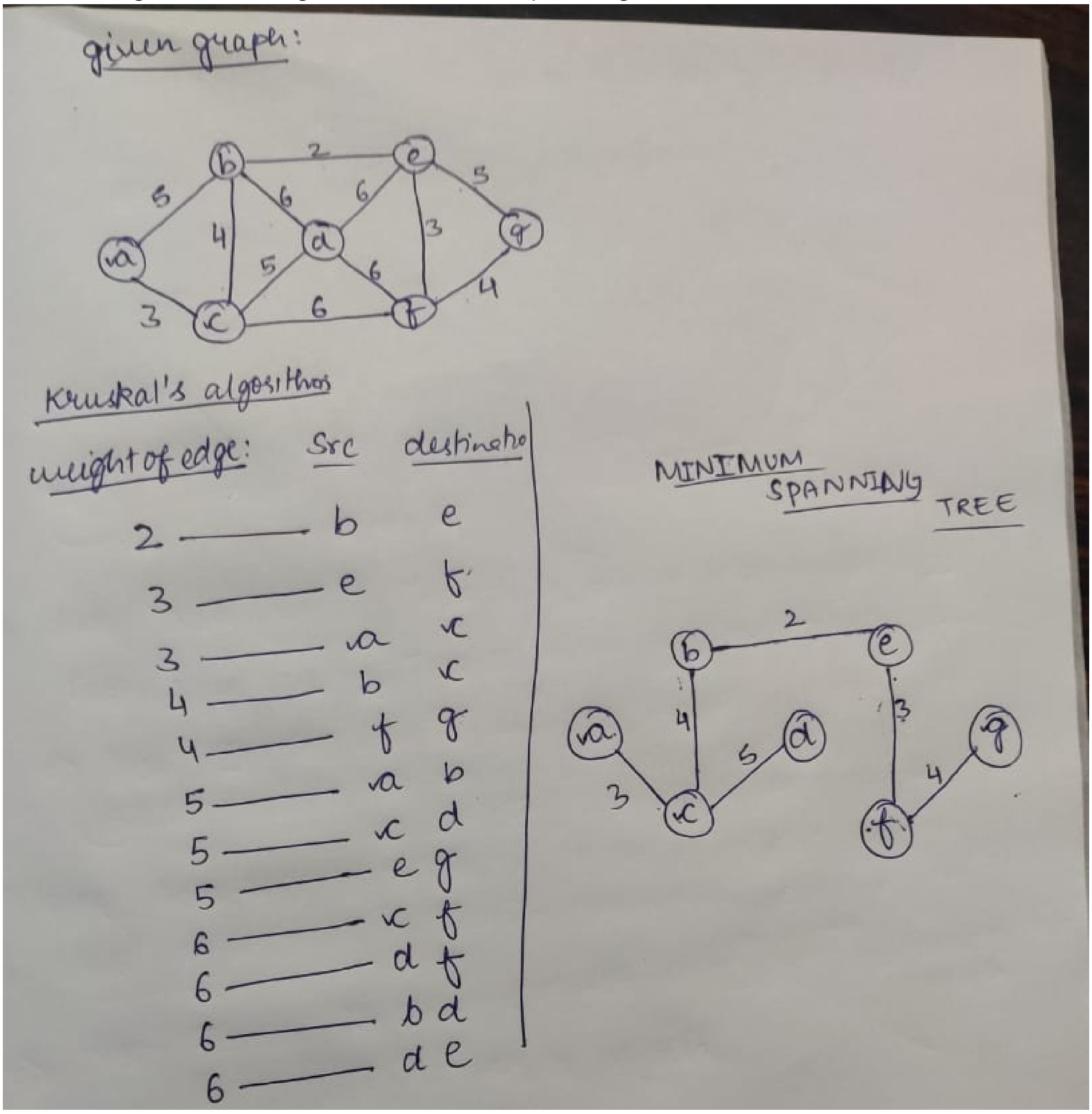
(A) (b; e)(e; f)(a; c)(b; c)(f; g)(c; d) (B) (b; e)(e; f)(a; c)(f; g)(b; c)(c; d)

(C) (b; e)(a; c)(e; f)(b; c)(f; g)(c; d) **(D) (b; e)(e; f)(b; c)(a; c)(f; g)(c; d)**

sol)

Option (D)

Using kruskal's algorithm , we get our minimum spanning tree as follows:



order of adding edges =

1) (b;e)
2) (e;f) } can be swapped
3) (a;c)
4) (b;c) } can be swapped
5) (f;g)
6) (c;d)

(d) is the only option not following the mention order

=> $\boxed{(d)}$

---

17. Which one of the following is TRUE for any simple connected undirected graph with more than 2 vertices?

(A) No two vertices have the same degree.
(B) At least two vertices have the same degree
(C) At least three vertices have the same degree.
(D) All vertices have the same degree.

Sol) Option (B)
Let us consider a simple connected undirected graph of ‘n’ vertices.
As it is a simple graph, there will be no self loops .
So , the degree of each vertex will be lying in the range of 1 - (n-1)
but since we have ‘n’ vertices , at least 2 vertices will have the same degree for sure.

---

18. Suppose that the universe U of possible keys is {0, 1,..., n^2-1}. For a hash table of size n, what is the greatest number of distinct keys the table can hold with each of these collision resolution strategies?

(A) Chaining Solution
(B) Linear probing Solution
(C)Quadratic probing Solution

sol)
(A) Chaining solution: **n^2**
(B) Linear Probing Solution: **n**
(C) Quadratic Probing Solution: **n**

In the chaining solution , we use linked lists to store the values to be stored under same index . So here , all the elements belonging to universal set can be stored => n^2
For the remaining two solutions , we simply use arrays to implement that logic. so since the hash is of size ‘n’ , a maximum of ‘n’ distinct keys can be held by the table

---

19. You are given a hash table with n keys and m slots, with the simple uniform hashing assumption (each key is equally likely to be hashed into each slot). Collisions are resolved by chaining. What is the probability that the first slot ends up empty?

sol)

given:

- A hash table has 'n' keys & m slots
- A Collision is solved by chaining

As all keys are likely to be hashed,

Probability of one key to be hashed in first slot $= P(1) = \frac{1}{m}$

Probability of one key to NOT be hashed in first slot $= \overline{P(1)} = 1 - \frac{1}{m} = \frac{m-1}{m}$

~~Probability of all 'n' keys to NOT be hashed to first~~

$$\left(\text{Probability of 'n' keys to NOT be hashed in first slot}\right) = \overline{P(1)} \times \overline{P(1)} \times \cdots \text{(n times)} = \left(\overline{P(1)}\right)^n = \left(\frac{m-1}{m}\right)^n$$

$$\boxed{\therefore \text{Ans} = \left(\frac{m-1}{m}\right)^n}$$

---

20. Suppose we store n elements in an m-slot hash table using chaining, but we store each chain (set of elements hashing to the same slot) using an AVL tree instead of a linked list. Also suppose that m = n, so the load facor α = n/m =1

(A) What is the expected running time of insert, delete, and search in this hash table? Why? Assume simple uniform hashing.

(B) What is the worst-case running time of insert, delete, and search in this hash table? Why? (Do not assume simple uniform hashing.)

Sol)

(A) Since it is simple uniform hashing,
it means that all the n keys will get hashed into n unique slots . Since it is and AVL tree instead of linked list , we will be inserting one element each in each slot => insertion to the root node of AVL Tree.
So, for inserting , procedure is as follows:
step 1 : Find the slot where the element is to be hashed O(1)
step 2 : Insert it as the root node in that AVL Tree O(log 1)=0
expected running time for inserting = O(1)
It will be the same for deleting and searching too.
=> Inserting Running time = O(1)
=> Deleting Running time = O(1)
=> Searching Running time = O(1)

(B)

Since it is non uniform hashing,
let us look at the process for inserting.
Insert :
step 1 : Find the slot where the element is to be hashed O(1)
step 2 : Insert it in the AVL Tree of that respective slot O(log n)
So overall Insert worst case time = O(1 + log n)
Similar for delete and search processes too as worst case of both delete and search functions is also O(log n)
=>Insert worst case = O(1 + log n)
=> Delete Worst Case = O(1 + log n)
=> Search Worst Case = O(1 + log n)

---

21. A hash table guarantees constant lookup time. True/false? explain.

sol)
False.
In a case where there is no collision possible , we can have a constant lookup time of O(1).
In every other case , where we have collisions , the lookup time doesn’t remain to be O(1).
In the worst case scenario,lookup time has a time-complexity of o(n) , when all the hash slots are filled.

---

22. A non-uniform hash function is expected to produce worse performance for a hash table than a uniform hash function. True/false? Explain.

sol)
True.
for uniform hash function , time complexity is O(1 + α) , where α(= n/m) is the load factor.
and for non-uniform hash function , time complexity is O(1 + n).

Always, α is less than or equal to 1, where as , n >= 1
=> α is always less than or equal to n
=> **O(1 + α) < = O(1 + n)**

---

23. Consider a hash table of size seven, with starting index zero, and a hash function as

follows. h(x) = (3x + 4) mod 7 Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1; 3; 8; 10 is inserted into an Array
using closed hashing? Note that denotes an empty location in an Array.

(A) 8, 3, 1, −, −, −, 10

(C) 1, 3, −, 10, −, −, 3

(B) 1, 8, 10, −, −, −, 3

(C) 1, 10, 8, −, −, −, 3

Sol)
option(B)
Assume linear probing solution for collision

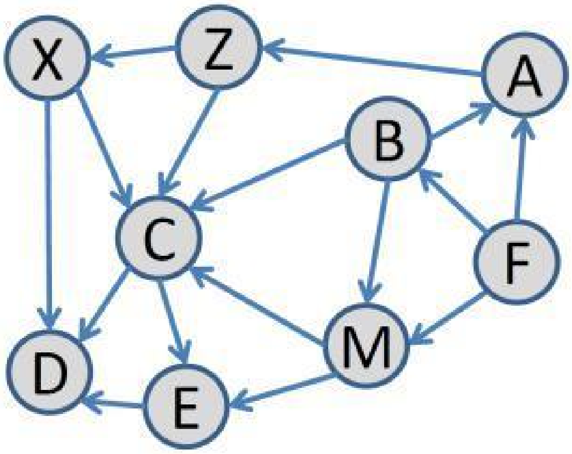
	0	1	2	3	4	5	6

Insert 1	1	-	-	-	-	-	-
insert 3	1	-	-	-	-	-	3
insert 8	1	8	-	-	-	-	3
insert 10	1	8	10	-	-	-	3

thus , option (b) is the correct answer

---

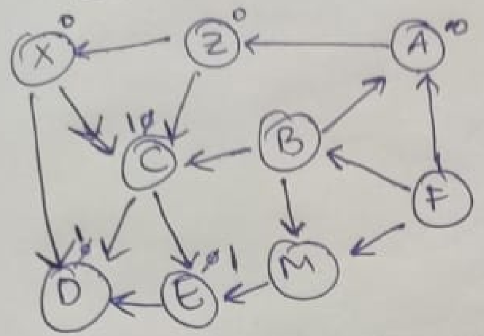
24. How many cycles does the following directed graph contain?



Sol) 0 (ZERO)
We check using stack if there is a cycle possible starting from vertex A and ending at A . We check the same for all vertices .

First let us mark all elements as -1, ~~iso~~ implying they are NOT visited.

-1 = unvisited
0 = visited & in stack
1 = visited & popped out of stack



start BFS traversal with A

element / parent

start BFS traversal with A

visited:

A Z X C D E

stack
E
D
C
X
Z
A

stack

pop E, D, C, X, Z, A

element	parent
A	
Z	A
C	Z
E	C
D	E

visited

A Z X C D E

empty

⇒ No cycle possible with A

similarly try starting with B, C, D, E, F, X, Z to find that no cycle can be formed with any of those vertices

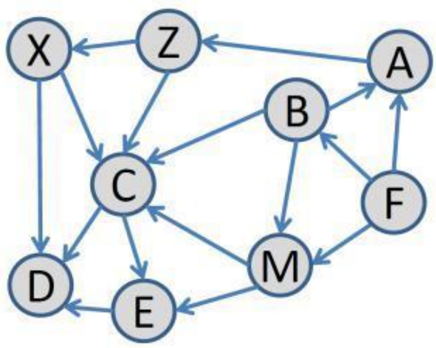
$\boxed{\therefore \text{ No of cycles the the given graph} = 0}$

---

25. For the same graph, write down all paths from node A to node D?

Sol)

Given graph is as follows:



A total of 5 paths are possible :

Path 1: A -> Z -> X -> D
Path 2: A -> Z -> C -> D
Path 3 : A -> Z -> C -> E -> D
Path 4 : A -> Z -> X-> C -> D
Path 5: A -> Z-> X -> C -> E -> D

---

26. Let G be a directed graph with a finite number of nodes. Are the following statements TRUE or FALSE?

(a) If G is directed acyclic, then there is a vertex with no incoming edges.

(b) If G is directed acyclic, then there is a vertex with no outgoing edges

Sol)

(A) True
(B) True

END
