GATE

CRASH COURSE

CS & IT

Algorithms

Graph Traversals-2
(Lecture 11)

By - Aditya sir

Topics to be Covered

1.
2. Graph Traversals
3. Practice Questions
4.

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored working professions in field of Data Science and Analytics
11. Have been mentoring GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Telegram Link for Aditya Jain sir:

https://t.me/AdityaSir_PW

Time Complexity of both DFS & BFS:

1) Adjacency Matrix: $O(n^2)$

2) Adjacency List: $O(n+e)$

$G(V, E)$

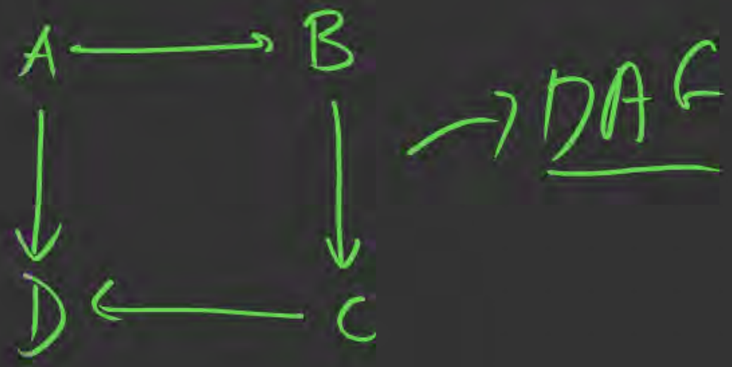
$|V| = n$

$|E| = e$

Applications of DFS:

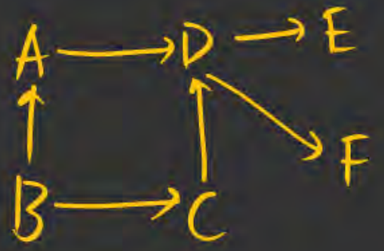
1) DAG → Topological Sort
2) Connected Components (CC)
3) Strongly Connected Components (SCC)
3) Bi-Connected Components (BCC)
4) Cut vertex / Articulation point

DAG → Directed Aaydic Graph :-



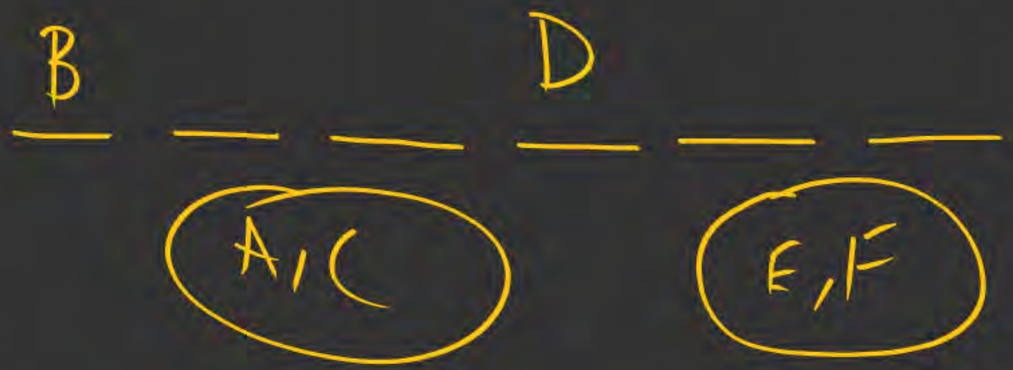
→ DAG

Topological Sort/ordering



B → Source

E, F → Sink



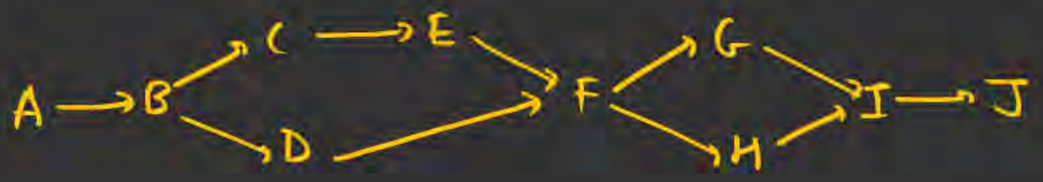
1) BACDEF

2) BCADEF

3) BACDFE

4) BCADFE

(a)



A B F I J

CED

G, H

{ CED, CDE, DCE }

{ GH, HG }

3 X 2 = 6

Topological orderings

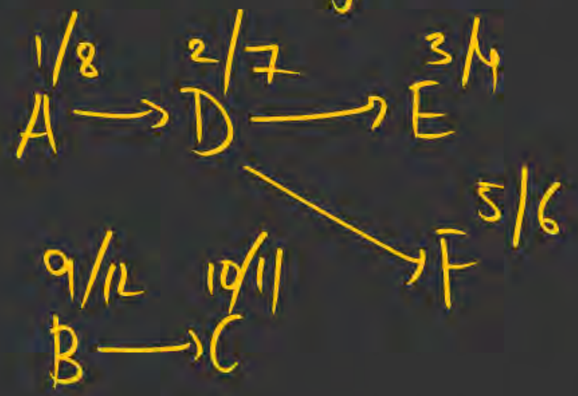
Topological Sort Algo → using DFS:

1) Apply DFS starting at any Node. ✓

2) Arrange nodes in descending / desc order of Finishing times.

eg:-

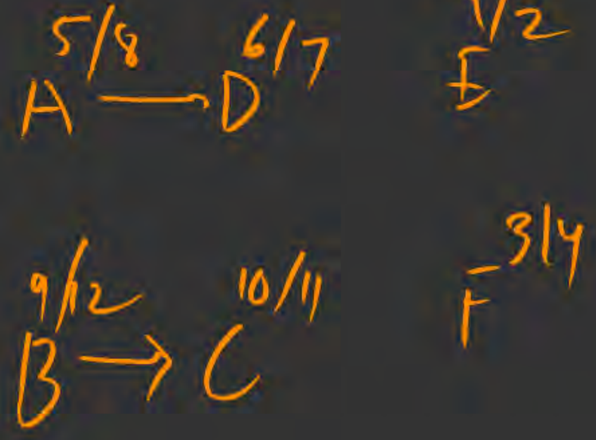
DFS starting at A



A → 8
B → 12
C → 11
D → 7
E → 4
F → 6

BCADFE

DFS at E



BCADFE

PYQ:

Undirected : P, Q, R

d(P) = 5 f(P) = 12

d(Q) = 6 f(Q) = 10

d(R) = 14 F(R) = 18

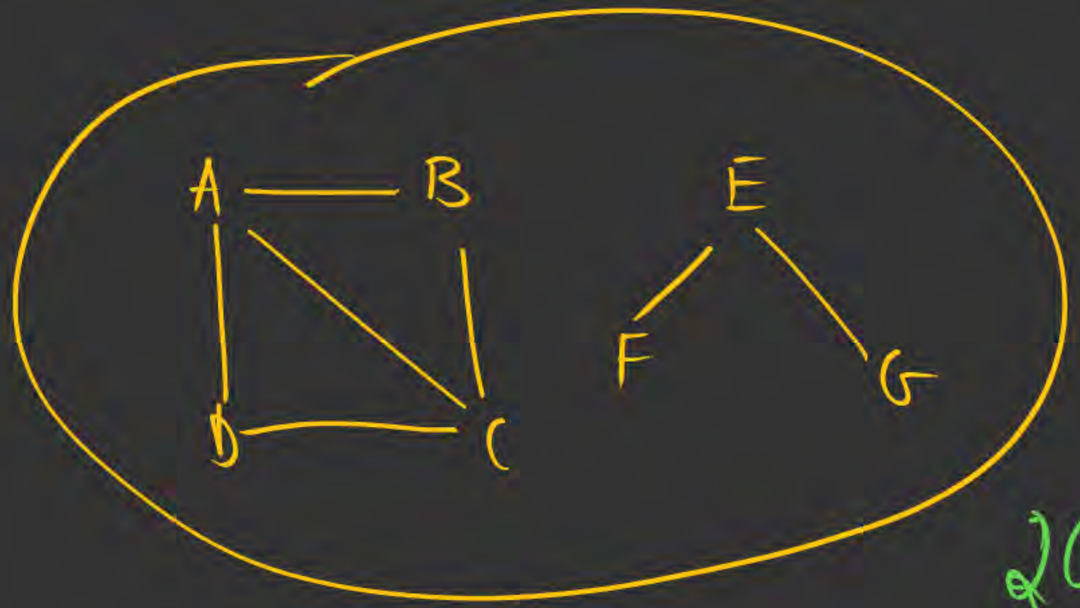
A) 1 connected component ✗

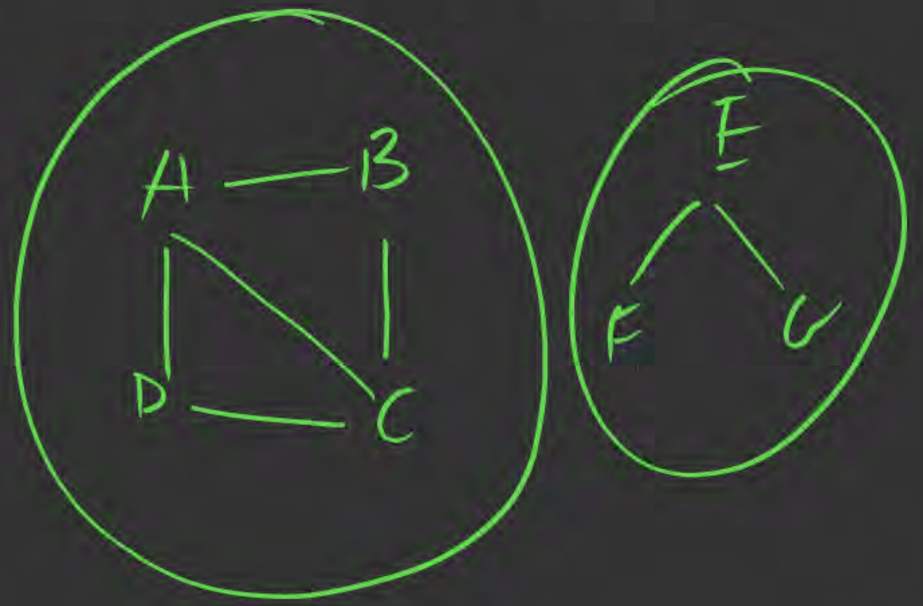
B) 2 CC, P — R ✗

C) 2 CC, Q — R ✗

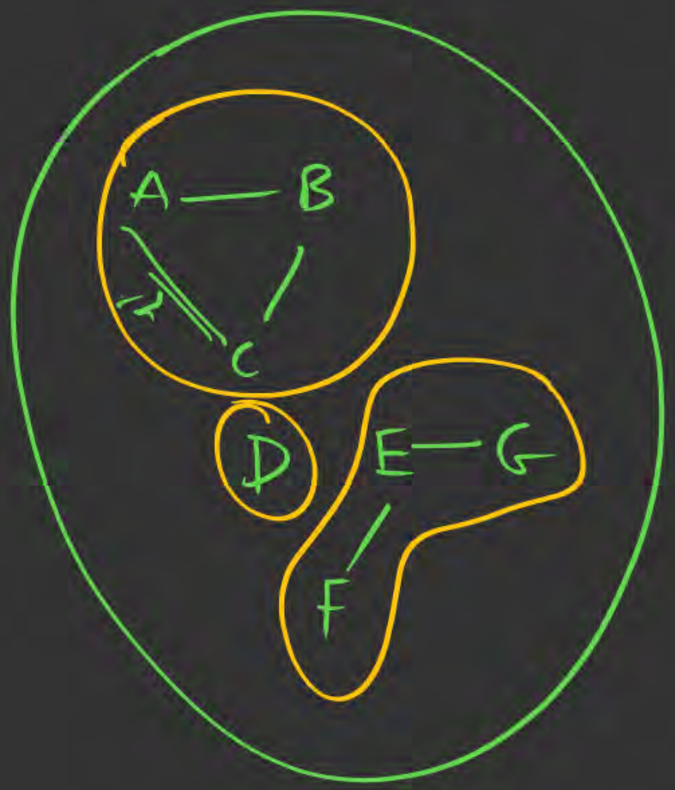
D) 2 CC, P — Q ✓

① Connected Components ⟶ (undirected graph)



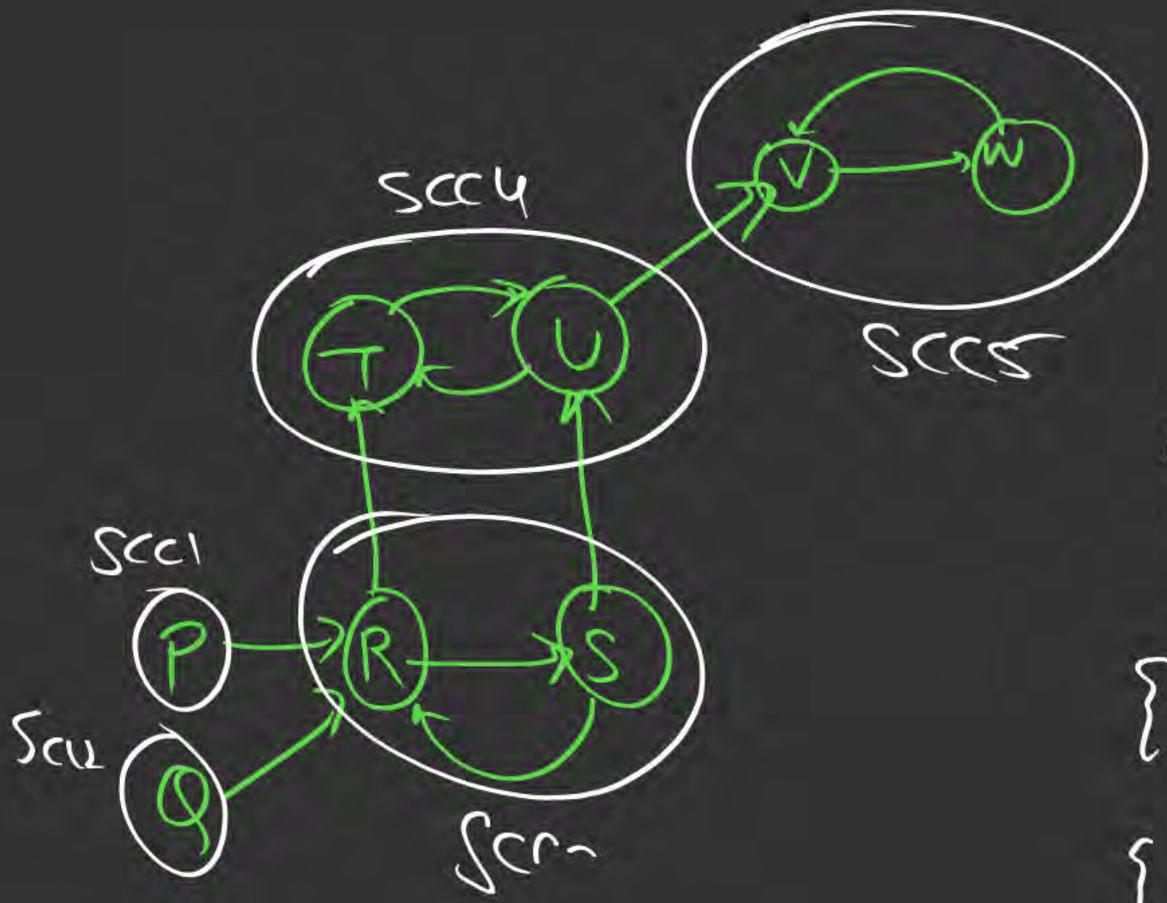


2CC



3 CC

② Strongly CC ⟶ Directed



{P}

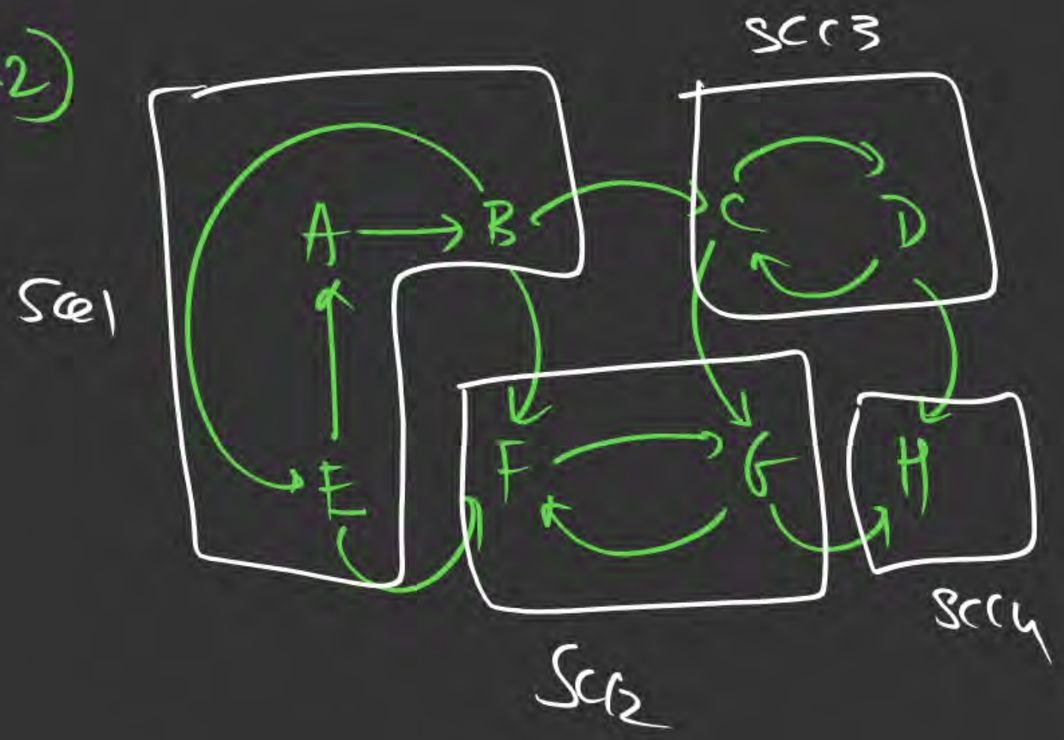
{Q}

{R,S}

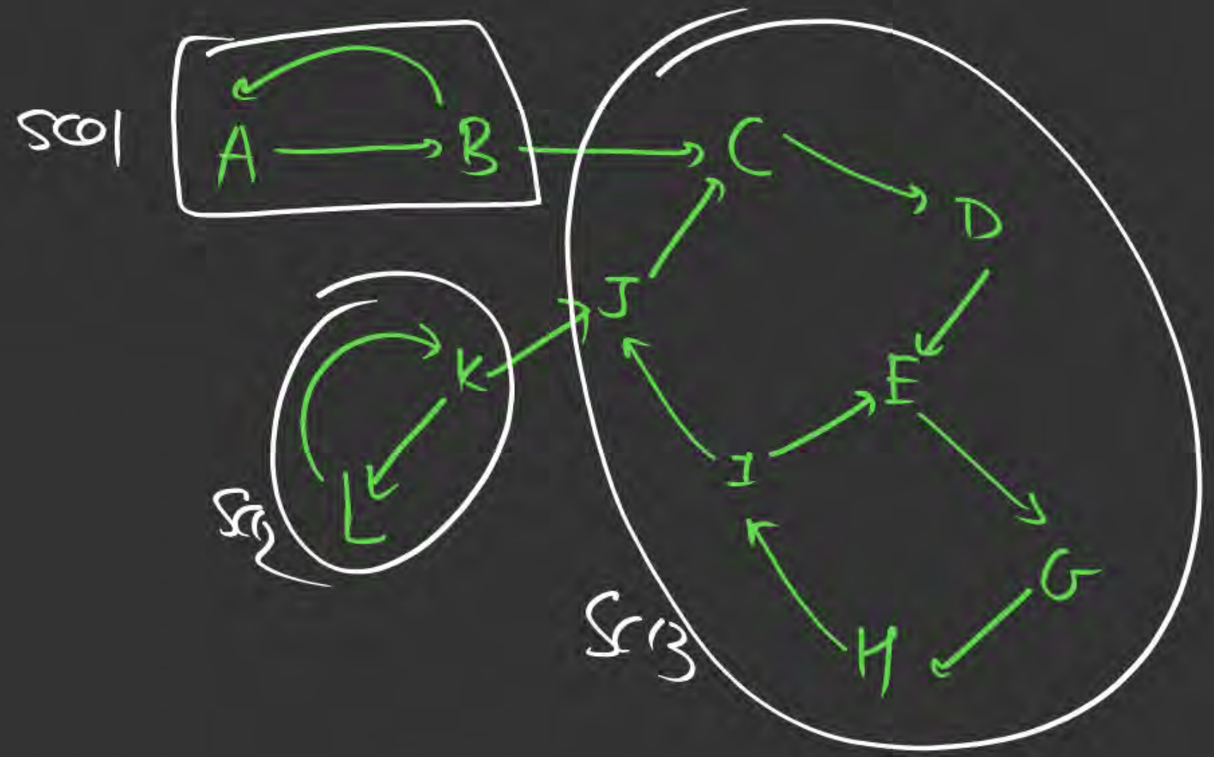
{T,U}

{V,W}

(Q.2)



(Q.3)



Topic : Strongly Connected Components

Property 1: Every directed graph is a D.A. G of its strongly connected components.

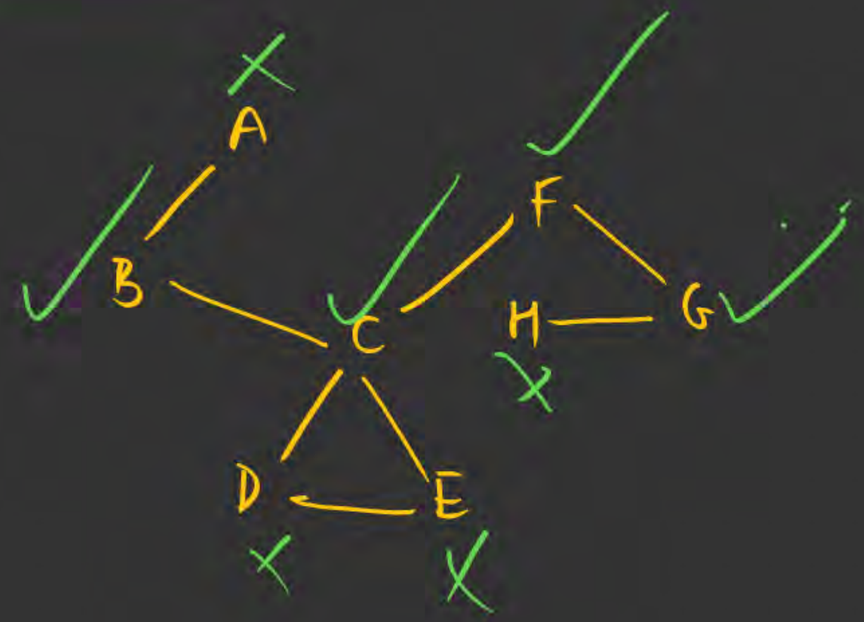
Property 2 : Let C and C be distinct strongly connected components in directed graph G = (V, E), let u, v, $\in$ C and u^1, $v^1 \in C^1$, suppose that there is a path $u \sim u^1$ in G, then there cannot also be a path $v^1 \sim v$ in G.

Property 3 : If 'C' and 'C^1' are strongly connected components of, and there is an edge from a node in C to a node in C^1, then the highest ~~post number~~ in C is bigger than the highest ~~post numb~~er in C^1.

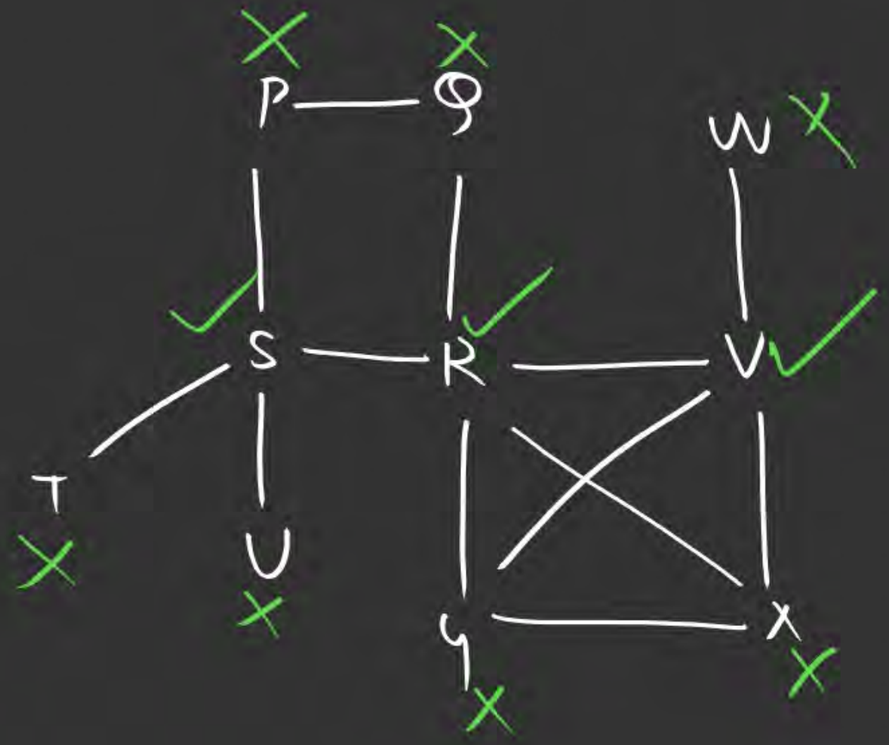
Finishing time

Finishing time

④ Articulation Point / Cut vertex

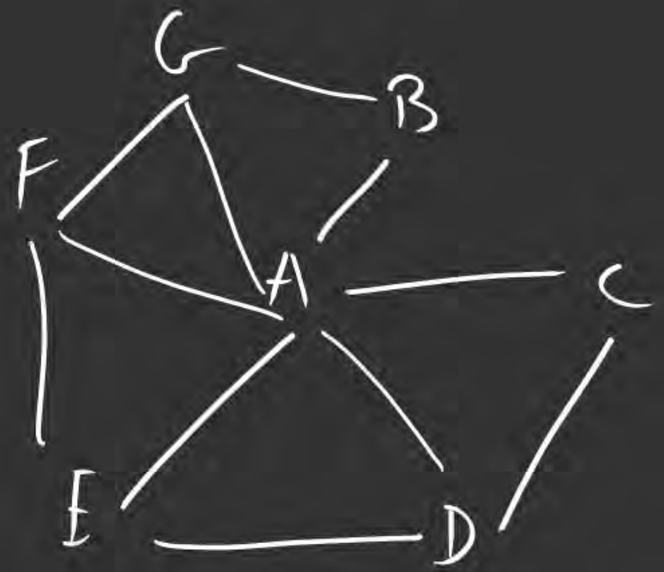


B, C, F, G → Articulation Points



Articulation points → S, R, V

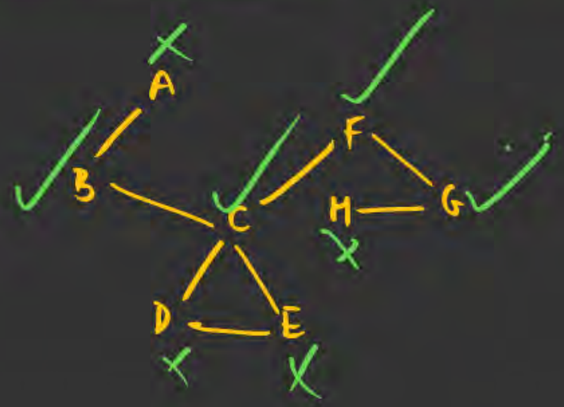
(Q.3)



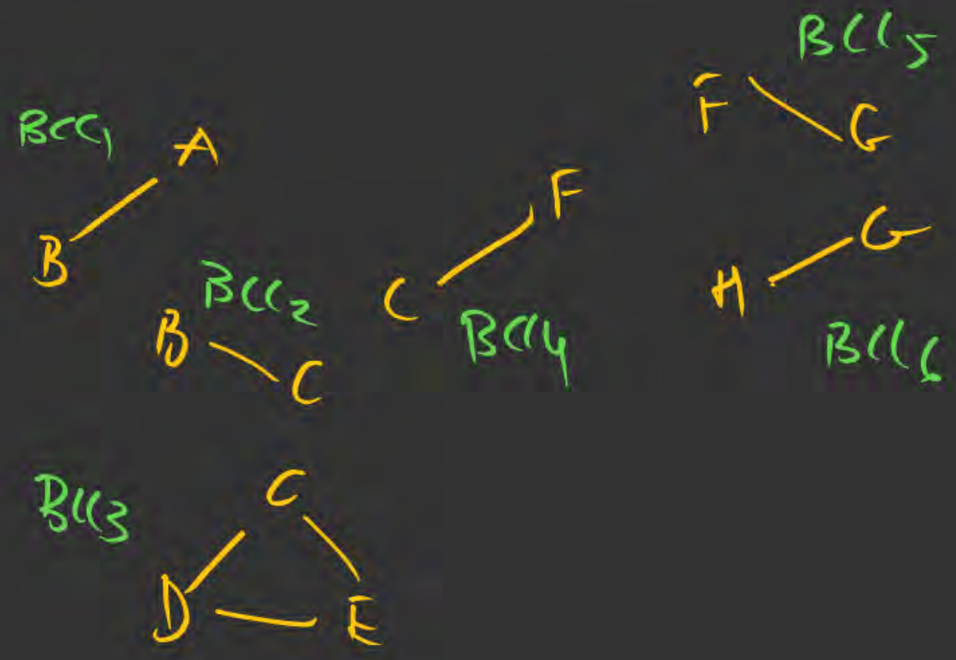
O Articulation Points

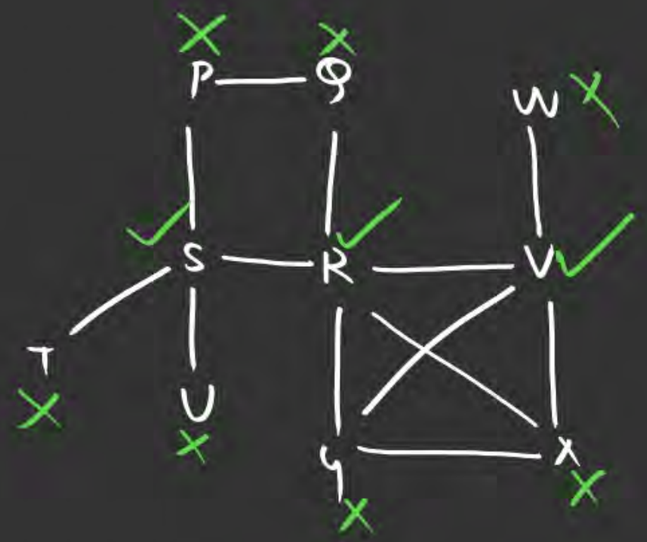
Bi-Connected Components

④ Articulation Point / Cut vertex

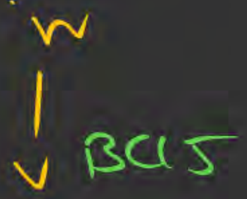


B, C, F, G → Articulation Points



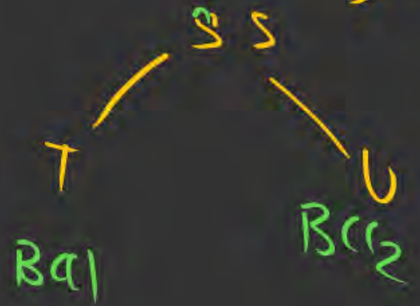


Articulation points → S, R, V



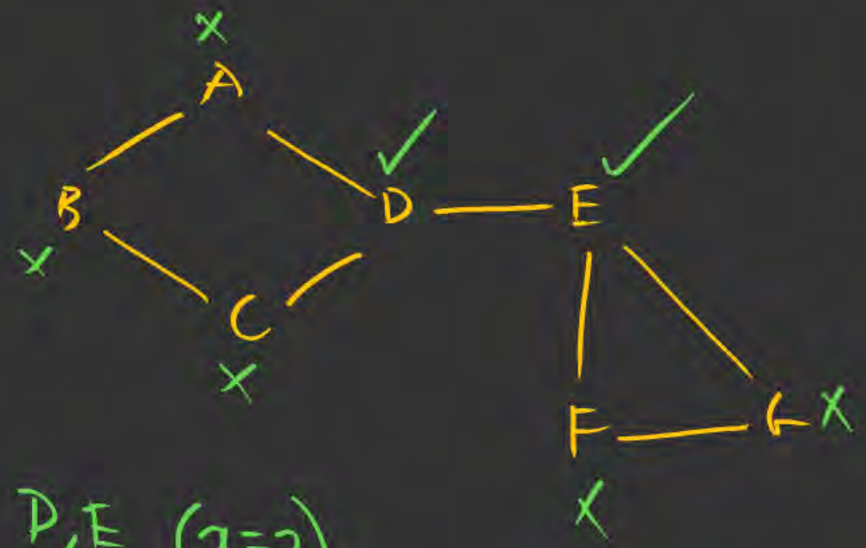
BCC5

5 BCCs



BCC1 BCC2

BCC4



APs: D, E ($x=2$)

$y = 3$

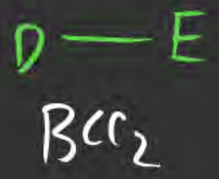
$x^y = 2^3 = 8$

APs → x

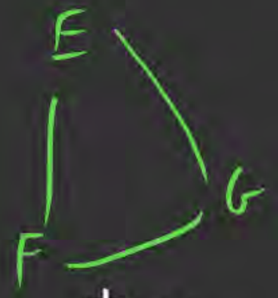
BCCs → y

$(x)^y = ?$

BCC1

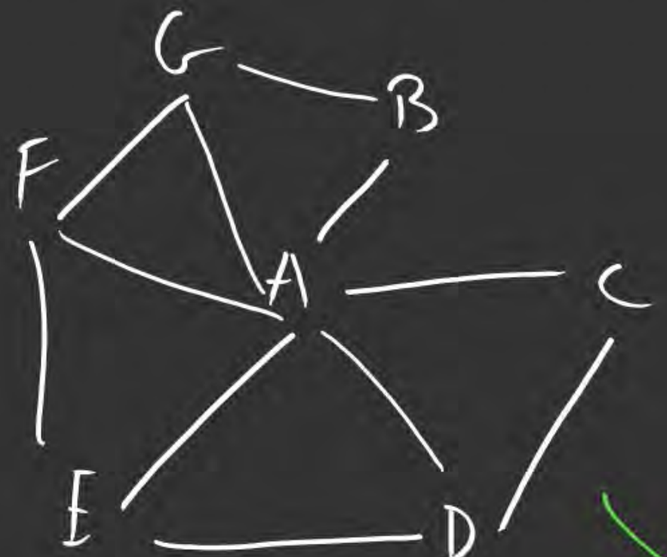


BCC2



BCC3

(Q.3)



O Articulation Points

Bi-Connected graph (1 Bcc)

Question

PYQ

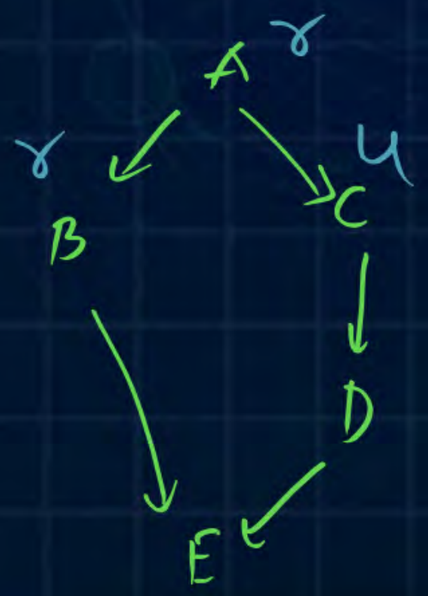
#Q. Consider and undirected graph (unweighted). If BFS of G is done from a node 'r' let d(r, u) and d(r, v) be the lengths of the shortest paths from r to u and v. If 'u' is visited before 'v', during the traversal, then which is true?

A. $d(r, u) < d(r, v)$

B. $d(r, u) > d(r, v)$

C. $d(r, u) \leq d(r, v)$ ✓

D. None



$d(r,u) = d(r,v)$

r u v

A B C E D

r

A C B D E

u v

$d(r,u) < d(r,v)$

$1 < 2$

Question

#Q. In a DF-traversal of a graph 'a' with n-vertices, 'k' edges are marked as tree edges, the number of connected components of 'G' is

for this eg:-

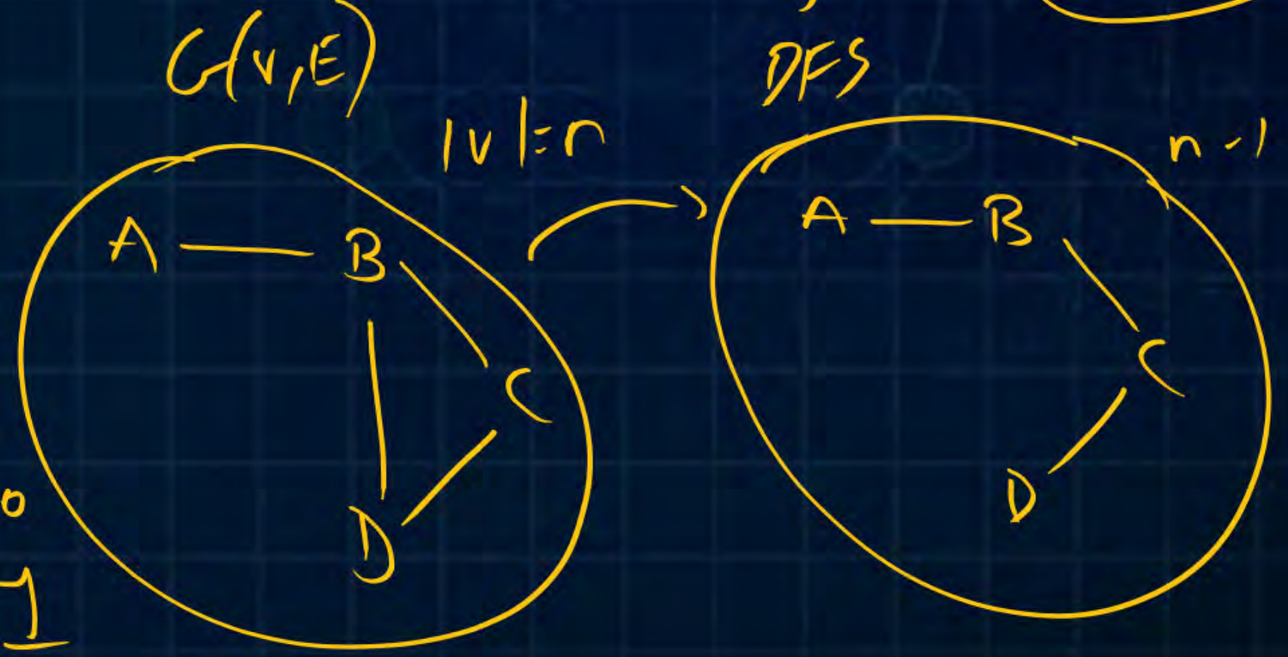
$K = n-1$

A K → $n-1$ ✗

B K + 1 → $n-1+1 = n$ ✗

C (n – k – 1) → $n-(n-1)-1$
$= n-n+1-1 = 0$ ✗

D n – k
↳ $n-(n-1)$
$= n-n+1 = 1$ ✓



Question

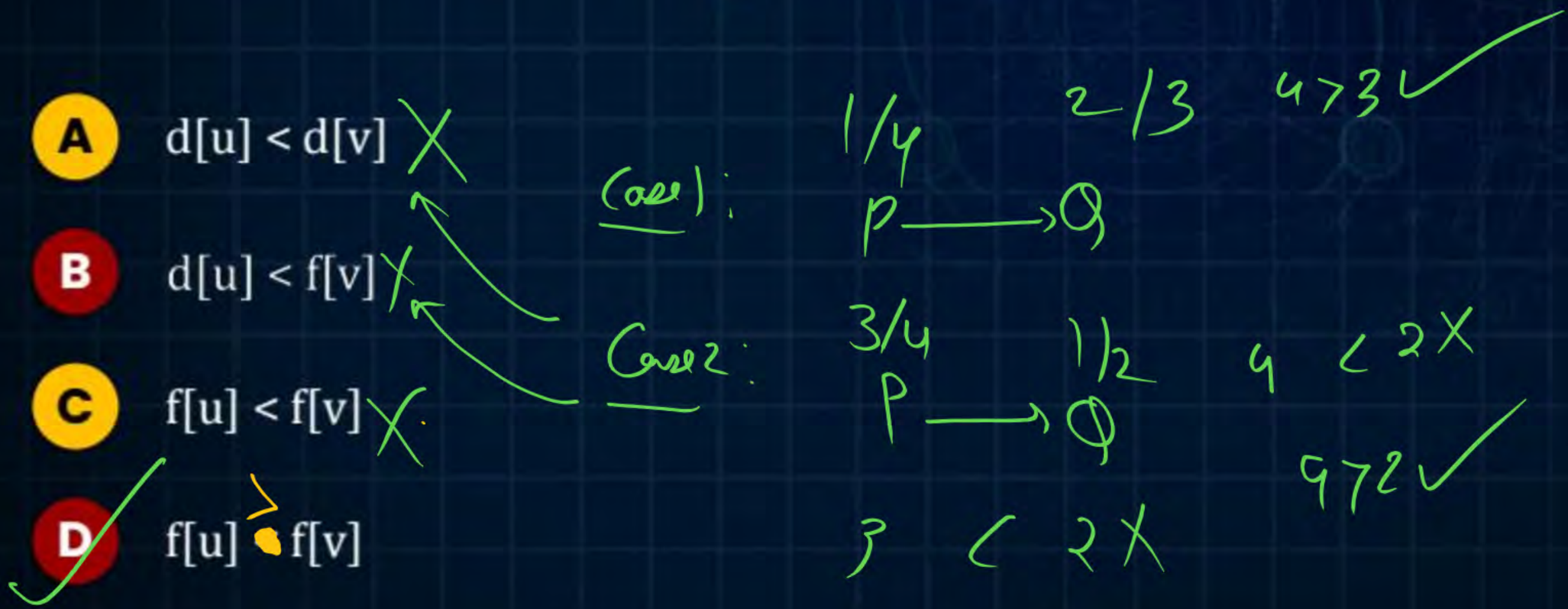
#Q.DFS is performed on a directed acyclic graph. D(u) is discovery time and f(u) is finishing time. Which is true for all edges (u, v) in the graph?

A. d[u] < d[v]

B. d[u] < f[v]

C. f[u] < f[v]

D. f[u] > f[v]



Question

#Q. Consider and undirected graph (unweighted). If BFS of G is done 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. If 'u' is visited before 'v', during the traversal, then which is true?

A $d(r, u) < d(r, v)$

B $d(r, u) > d(r, v)$

C $d(r, u) \leq d(r, v)$

D None

Question

#Q. In a DF-traversal of a graph ‘a’ with n-vertices, ‘k’ edges are marked as free edges, the number of connected components of ‘G’ is

A K

B K + 1

C (n – k – 1)

D n – k

Question

#Q. DFS is performed on a directed a cyclic graph. D(u) is discovery time and f(u) is finishing time. Which is true for all edges (u, v) in the graph?

A. $d[u] < d[v]$

B. $d[u] < f[v]$

C. $f[u] < f[v]$

D. $f[u] < f[v]$

Question

#Q. Consider the following functions from positive integers to real number:

$$f_1(n) = 2^{100}$$

$$f_2(n) = n$$

$$f_3(n) = n\log_2 n$$

$$f_4(n) = \frac{2^{100}}{n}$$

The correct arrangement of the above functions in increasing order of asymptotic complexity is:

A f_3, f_4, f_1, f_2

B f_4, f_1, f_2, f_3

C f_1, f_4, f_2, f_3

D f_4, f_1, f_3, f_2

Question

#Q.Merging 4 sorted files having 200,100, 250, 150 records will take how many comparisons to be merged into a single sorted file, if 2 files are merged at a time?

Question

#Q. Consider the following functions:

$f_1 = 2^{2n}$

$f_2 = n!$

$f_3 = 4^n$

$f_4 = 2^n$

What is the correct Decreasing order of above functions?

A $f_1\ f_4\ f_3\ f_2$

B $f_4\ f_2\ f_3\ f_1$

C $f_1\ f_2\ f_3\ f_4$

D $f_4\ f_3\ f_2\ f_1$

Question

#Q.Assume that there are 8n sorted list of size n/4 then what is the time complexity of merging them into single sorted list?

A $\theta(n^2 \log n)$

B $(\log n)$

C $\theta(n \log n)$

D $\theta(n^2)$

Question

#Q.Sort the functions in ascending order of asymptotic(big-O) complexity.

$f_1(n) = n$, $f_2(n) = (0.5)^n$, $f_3(n) = n\log n$, $f^4(n) = 5000$, $f_5(n) = (\log n)\log n$

A $f_4(n), f_2(n), f_1(n), f_5(n), f_3(n)$

B $f_2(n), f_1(n), f_4(n), f_5(n), f_3(n)$

C $f_2(n), f_4(n), f_1(n), f_5(n), f_3(n)$

D $f_1(n), f_5(n), f_4(n), f_3(n), f_2(n)$

Question

#Q. Consider the following code.

```
main( )
    {
        i=1;
        while(i <= n)
        {
            i=10*i;
        }
    }
```

What is the highest asymptotic worst case time complexity of above code fragment?

A $O(n^2)$

B $O(\sqrt{n})$

C $O(n)$

D $O(\log n)$

Question

#Q. Consider a list which contains np sorted array each of size n/p and is merged using merge sort, then what is the tightest upper bound worst case complexity?

A $O(np^2 lognp)$

B $O(n^2 logn)$

C $O(n^2 lognp)$

D None of these

Question

#Q. What is the time complexity of the following code ?

```
for (a = 0; a <= n ; a =a*2)
{
  for (b = 0; b < 100 ; b = b +2)
      {
      for (c = 1; c < 8*n; c ++)
          {
              print("AJ Sir")
          }
      }
}
```

A $O(n^3)$

B $O(n^2)$

C O(logn)

D None of These

Question

#Q. $f(n) = \sum_{i=1}^{n} i^3$ then choices for f(n):

I. $\theta(n^3)$
II. $\theta(n^5)$
III. $O(n^5)$
IV. $\Omega(n^3)$

A I

B II

C III

D IV

Question

#Q. Consider a list which contains np sorted array each of size n/p and is merged using merge sort, then what is the tightest upper bound worst case complexity?

A $O(np^2 lognp)$

B $O(n^2 logn)$

C $O(n^2 lognp)$

D None of these

Question

#Q. Consider the two input array a_1 and a_2 with elements [12345] and [54321]. If quick sort program is used to sort numbers into ascending order the time taken by array a_1 and a_2 is time t_1 and t_2 respectively. Then, what is the relation between t_1 and t_2.

A $T_1 > T_2$

B $T_1 < T_2$

C $T_1 = T_2$

D $T_2 = T_1 \log T_1$

Question

#Q. The Flyod-Warshall algorithm for all pairs shortest paths computation is based on

A Greedy method

B Divide and Conquer

C Dynamic Programming

D Heap algorithm

Question

#Q. Suppose that there are 3 programs X_1, X_2 and X_3 having time complexities $f_1(n)$, $f_2(n)$ and $f_3(n)$ respectively. Such that $f_1(n)$ is $O(f_2(n))$, $f_2(n)$ is $O(f_1(n))$, $f_1(n)$ is $O(f_3(n))$ and $f_3(n)$ is not $O(f_1(n))$. Then which one of the statements is true from the following statements?

A X_3 is always faster than X_1 and X_2 for very large size inputs

B X_1 is faster than X_2 and X_3 for very large inputs

C X_3 is slower than X_1 and X_2 for very large input

D X_2 is faster than X_1 and X_3 for very large size inputs

Question

#Q. Consider a list which contains np sorted array each of size n/p and is merged using merge sort, then what is the tightest upper bound worst case complexity?

A $O(np^2 lognp)$

B $O(n^2 logn)$

C $O(n^2 lognp)$

D None of these

Thank

THANK

Keep Hustling!