End semester Examination (Monsoon 2023-24)

Department of Computer Science & Engineering, IIT (ISM), Dhanbad

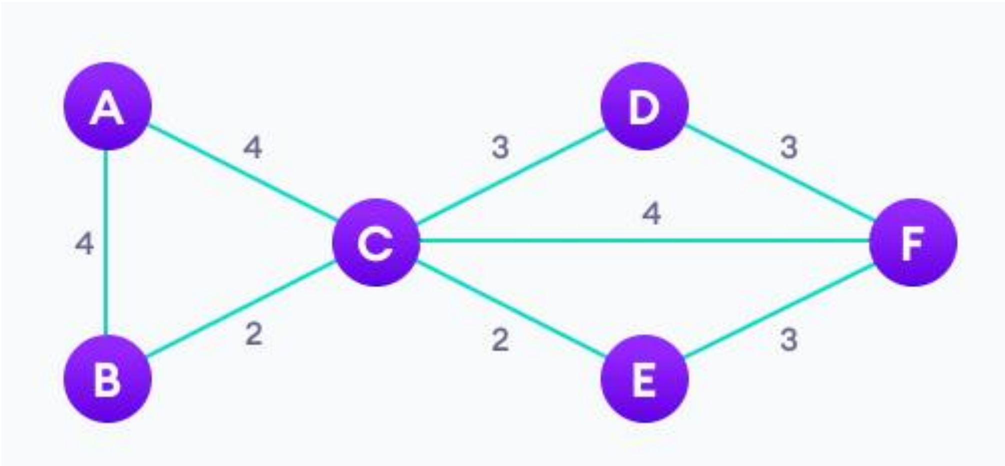
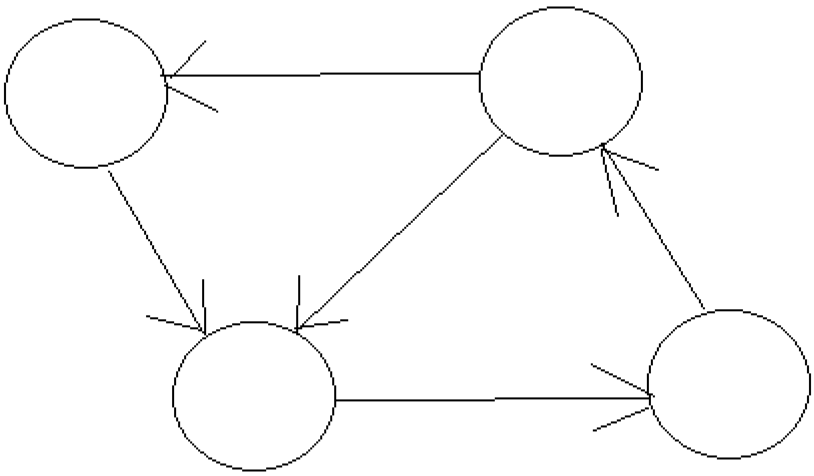
Discipline: M.Tech. (CSE) & Research Scholar

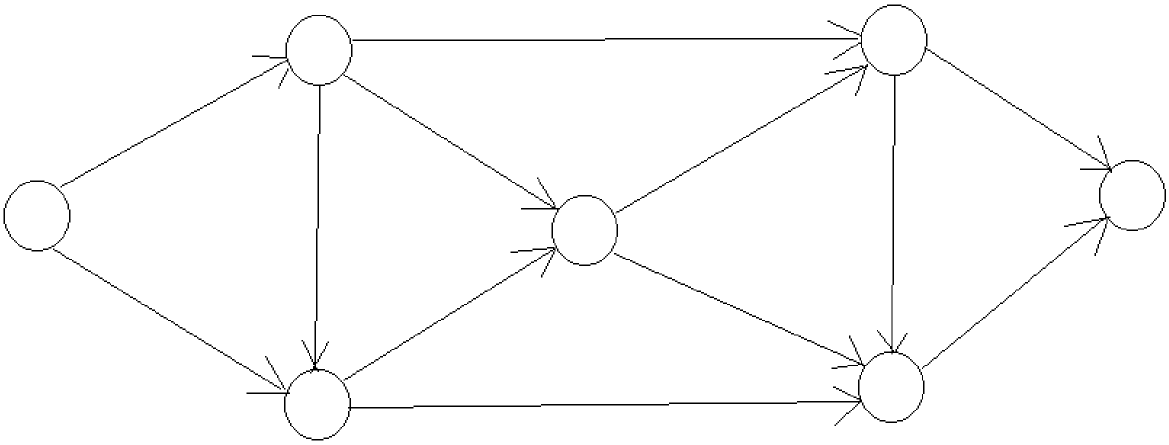
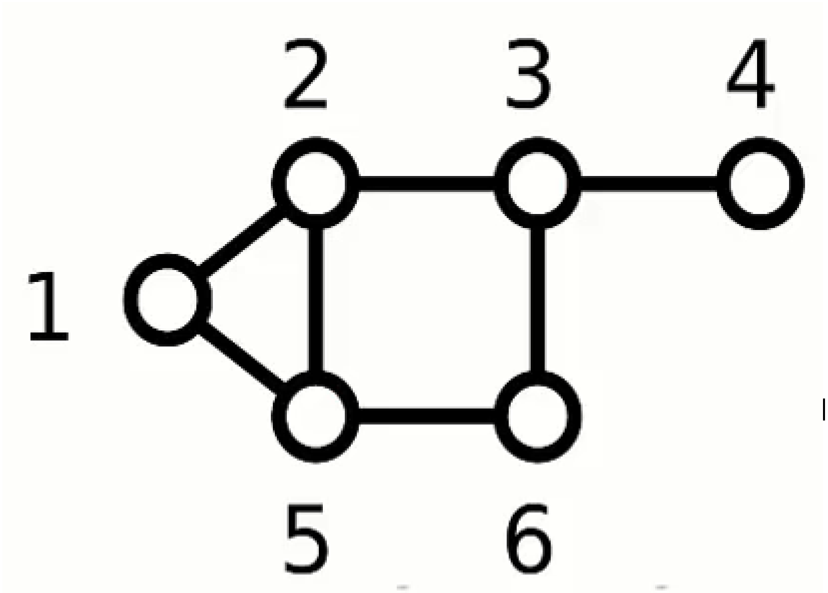
Subject: Algorithmic Graph Theory (CSC503) **Time: 3** hours, Marks: 100

Instructions: *Answer all questions*

Q. No		Marks
1	a. Prove that for every graph G, X (G) <= V(G) +1. (Here X(G) is the chromatic number of G) b. Prove that G is bipartite iff the chromatic number of G is 2.	5 5
2	a. The Prufer sequence S of a labelled tree T is S= {1, 7, 6, 6, 1}. Draw the tree T describing all the intermediate steps pictorially during the process. b. Write the algorithm of the same.	5 5
3	a. Prove by method of induction that a graph G is a tree iff G is acyclic and the number of edges m in G is equal to n-1, where n is the number of vertices in G. b. Deduce that clique decision problem is NP-Hard using Satisfiability problem.	5 5
4	a. Prove that for a simple graph G with n number of vertices (n >= 4), and E number of edges and genus g satisfies $$g \geqslant \lceil \tfrac{1}{6}(\lvert E\rvert - 3n) + 1 \rceil$$ b. “A necessary and sufficient condition for a graph G to be planar is that for every circuit C of G the auxiliary graph $G^+(C)$ is bipartite”, prove it.	5 5

5	a. Suppose want to schedule some final exams for CS courses with following call numbers: CSE001, CSE002, CSE003, CSE004, CSE005, CSE006, CSE007, CSE008 and also suppose also that there are no common students in the following pairs of courses because of prerequisites: CSE001 - CSE002 CSE001 - CSE004 CSE001 - CSE003, CSE002- CSE003 CSE001- CSE007, CSE002 - CSE007 CSE001 - CSE008, CSE002 - CSE008, CSE003 - CSE008 CSE001 - CSE005, CSE002 - CSE005, CSE004 - CSE005, CSE001 - CSE006, CSE002 - CSE006, CSE004 - CSE006, CSE005 - CSE006 How many exam slots are necessary to schedule exams? b. Illustrate with an example that any graph is a subgraph of a r-regular graph.	6 4
6	a. If a graph G is maximal planar graph with *n* vertices (*n>=3*) and *m* edges then show that *m = 3n – 6.* b. For a polyhedron with *N* number of vertices, *E* number of edges and *F number of faces*, prove by method of induction that *N – E + F = 2.*	5 5
7	a. Why running time measurement is important to study? Mention some of the factors affecting the running time of a program. b. Prove that every planar graph has a dual.	2+3 5
8	a. Discuss the merits and demits of Dijkstra’s Algorithm and Bellman-Ford Algorithm with examples.	5

	b. Use Prim's algorithm to find a minimum spanning tree that takes the following graph as input. 	5
9	a. Find the all pairs shortest paths among all vertices for the following graph using Floyd-Warshall Algorithm (showing all intermediate steps) 	4
	a. Prove that Peterson Graph is non-planar using Kuratowski's and Wagner's Theorem as well.	6

10	a. Find the maximum flow through the given network using Ford-Fulkerson algorithm.  b. Consider the following graph to find the minimum connected dominating set showing all the intermediate steps during the constructions.  	5 5

Ans1.a

<u>Theorem</u>: For every graph G, $\chi(G) \le \Delta(G)+1$

<u>Proof</u>: By induction on n (the # of vertices)

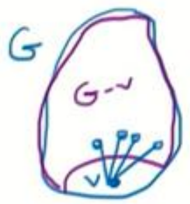
Basis: $n=1$ $\quad G = K_1$

$\chi(G) = 1$ ✓

$\Delta(G) = 0$

Ind. Hyp: Assume the result holds for every graph with $n-1$ vertices $(n \ge 2)$.

Let G be a graph on n vertices. Let $v \in V(G)$.



Now $G-v$ is a graph on $n-1$ vertices

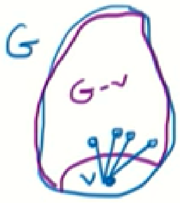
Ind Hyp $\Rightarrow$ $\chi(G-v) \le \Delta(G-v)+1$

So $G-v$ can be coloured with at most $\Delta(G-v)+1$ colours.

Note: $\deg_G(v) \le \Delta(G)$

The neighbours of v use up $\le \Delta(G)$ colours

Let G be a graph on n vertices. Let $v \in V(G)$.



Now $G-v$ is a graph on $n-1$ vertices

Ind Hyp $\Rightarrow \chi(G-v) \leq \Delta(G-v)+1$

So $G-v$ can be coloured with at most $\Delta(G-v)+1$ colours.

Note: $\deg_G(v) \leq \Delta(G)$

The neighbours of v use up $\leq \Delta(G)$ colours

Case 1: $\Delta(G) = \Delta(G-v)$

Then there is at least one colour of the $\Delta(G-v)+1 = \Delta(G)+1$ colours not used by the neighbours of v, so v can be coloured with that colour.

This gives a $(\Delta(G)+1)$-colouring of G, so $\chi(G) \leq \Delta(G)+1$ ✓

Case 2: $\Delta(G) \neq \Delta(G-v)$

Then $\Delta(G-v) < \Delta(G)$

Using a <u>new</u> colour for v we will have a

$(\Delta(G-v)+2)$-colouring of G

Since $\Delta(G-v)+2 \leq \Delta(G)+1$, we have $\chi(G) \leq \Delta(G)+1$ ✓

$\chi(G) \leq \Delta(G) + 1$

Note: $\left.\begin{array}{l} \chi(K_n) = n \\ \Delta(K_n) = n-1 \end{array}\right\}$ equality in the bound

$\left.\begin{array}{l} \chi(C_{2\ell+1}) = 3 \\ \Delta(C_{2\ell+1}) = 2 \end{array}\right\}$ equality in the bound

odd cycles and complete graphs are the only graphs that have equality in this bound

$\left.\begin{array}{l} G \neq K_n \\ G \neq C_{2\ell+1} \end{array}\right\} \Rightarrow \chi(G) \leq \Delta(G)$ Brook's Theorem

Xxxxxxxxxxxxxxxxxx

Ans 1,b

Theorem: $\chi(G) = 2 \iff G$ is bipartite (for non-trivial graphs)

(Recall: G is bipartite $\iff$ G has no odd cycle)

Proof:

$\Rightarrow$] Aim to show: $\chi(G) = 2 \Rightarrow G$ is bipartite

Contrapositive: G is not bipartite $\Rightarrow \chi(G) > 2$

Suppose G is not bipartite.

Then G has an odd cycle $C_{2\ell+1}$ and hence $\chi(G) \geq \chi(C_{2\ell+1}) = 3$ ✓

$\Leftarrow$] Aim to show: G is bipartite $\Rightarrow \chi(G) = 2$

Suppose G is bipartite. Then $V(G) = X \cup Y$ such that every edge of G is of the form $e = xy$ where $x \in X$ and $y \in Y$

Then colour all vertices of X colour 1 and all vertices of Y colour 2.

$\therefore \chi(G) = 2$

■

Xxxxxxxxxxxxxxxxxxxxx

Q2 a.

Prüfer sequence S of length $n-2$ using symbols $1,\dots,n$ $\longrightarrow$ labelled tree T

- $S=(a_1,a_2,\dots,a_{n-2})$ $\quad a_i \in \underbrace{\{1,\dots,n\}}_{L}$
- Find smallest element $x \in \{1,\dots,n\}$ $\quad x \notin S$

  ↳ join x to 1^{st} element of S

→ delete a_1 from S, delete x from $\underbrace{\{1,\dots,n\}}_{L}$

- Find smallest $y \in L \setminus \{x\}$ not in S

• $S = (a_1, a_2, \dots, a_{n-2})$ $\quad a_i \in \underbrace{\{1, \dots, n\}}_{L}$

• Find smallest element $x \in \{1, \dots, n\}$ $\quad x \notin S$

↳ join x to 1st element of S

→ delete a_1 from S, delete x from $\underbrace{\{1, \dots, n\}}_{L}$

• Find smallest $y \in L \setminus \{x\}$ not in $S \setminus \{a_1\}$

↳ join y to the 1st element of $S \setminus \{a_1\}$

→ delete a_2 from S
delete y from L

↳ join x to 1^{st} element of S

→ delete a_1 from S, delete x from $\underbrace{\{1,\dots,n\}}_{L}$

- Find smallest $y \in L \setminus \{x\}$ not in $S \setminus \{a_1\}$

  ↳ join y to the 1^{st} element of $S \setminus \{a_1\}$

  → delete a_2 from S
  delete y from L

- Continue until 2 items remain in L

  ↳ join these two via an edge

delete y from L y a_2

- Continue until 2 items remain in L
  - ↳ join these two via an edge

$S = (1, 7, 6, 6, 1)$ $\underline{\text{length}}\ 5 \Rightarrow \boxed{n=7}$

$L = \{1, 2, \dots, 7\}$

$x \in L \setminus S$ smallest $\rightarrow x = 2$

delete y from L y a_2

- Continue until 2 items remain in L
  ↳ join these two via an edge

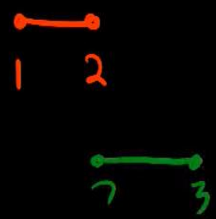
$S = (\cancel{1}, 7, 6, 6, 1)$ $\underline{\text{length}}\ 5 \Rightarrow \boxed{n=7}$

$L = \{1, \cancel{2}, \ldots, 7\}$

$x \in L \setminus S$ smallest $\to x = 2$

$S = (7, 6, 6, 1)$

$L = \{1, 3, 4, 5, 6, 7\}$



delete y from L y a2

- Continue until 2 items remain in L
  ↳ join these two via an edge

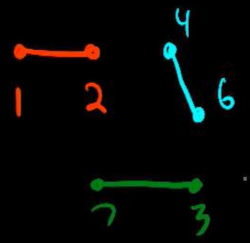
$S = (\cancel{4}, 7, 6, 6, 1)$ $\underline{\text{length}}\ 5 \Rightarrow \boxed{n=7}$

$L = \{1, \cancel{2}, \ldots\ldots, 7\}$

$x \in L \setminus S$ smallest $\rightarrow x = 2$

$S = (\cancel{7}, \cancel{6}, 6, 1)$

$L = \{1, \cancel{3}, \cancel{4}, 5, 6, 7\}$



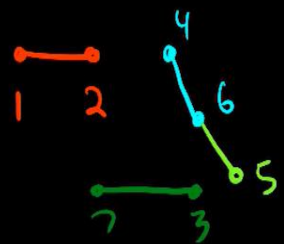
delete y from L y a_2

- Continue until 2 items remain in L
  ↳ join these two via an edge

$S = (\cancel{4}, 7, 6, 6, 1)$ length 5 $\Rightarrow$ $\boxed{n = 7}$

$L = \{1, \cancel{2}, \ldots, 7\}$

$x \in L \setminus S$ smallest $\rightarrow x = 2$



$S = (\cancel{7}, \cancel{6}, \cancel{6}, 1)$

$L = \{1, \cancel{3}, \cancel{4}, \cancel{5}, 6, 7\}$

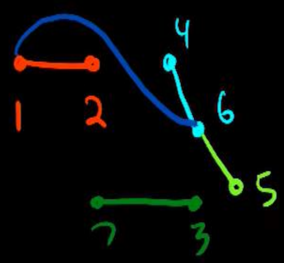
delete y from L   y   a2

- Continue until 2 items remain in L
  - ↳ join these two via an edge

$S = (\cancel{4}, 7, 6, 6, 1)$ length 5 $\Rightarrow$ $\boxed{n=7}$

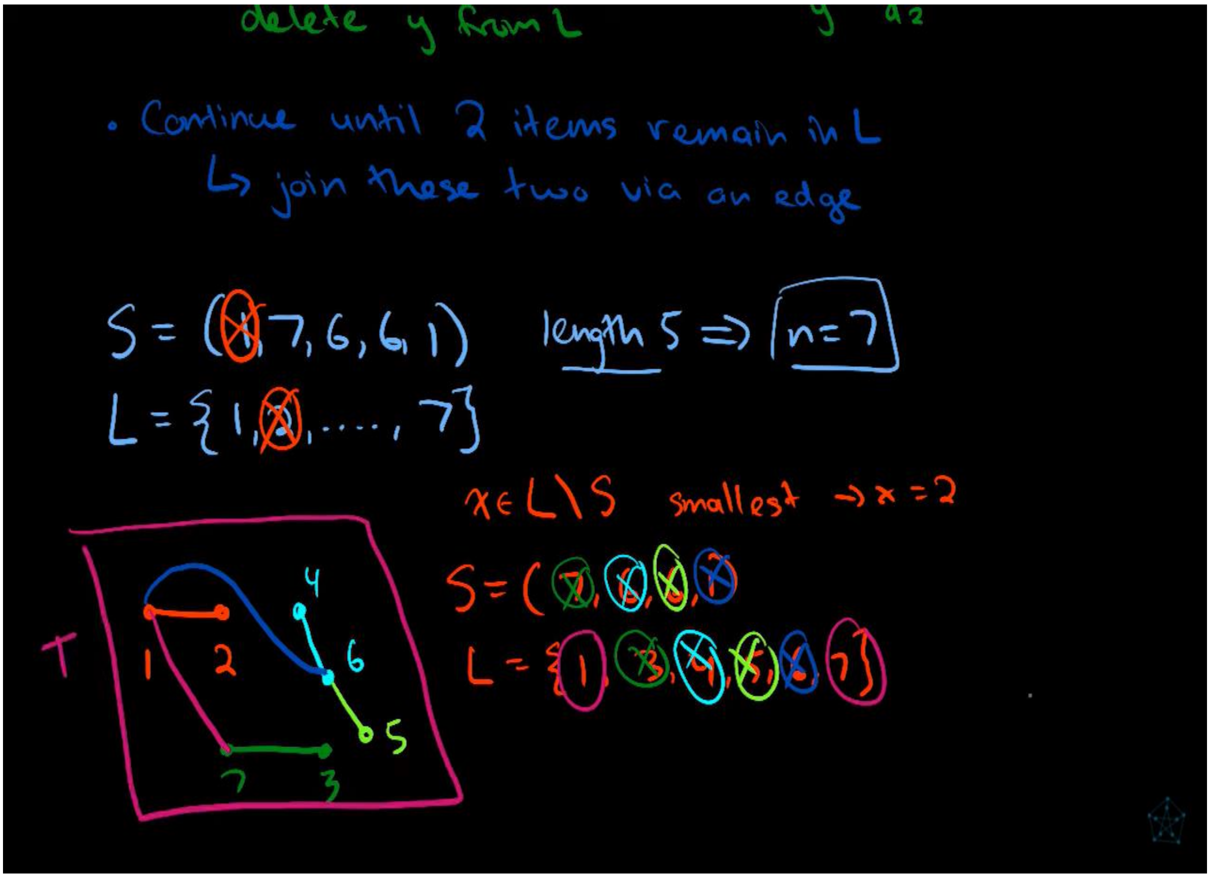
$L = \{1, \cancel{2}, \ldots, 7\}$

$x \in L \setminus S$ smallest $\rightarrow x = 2$



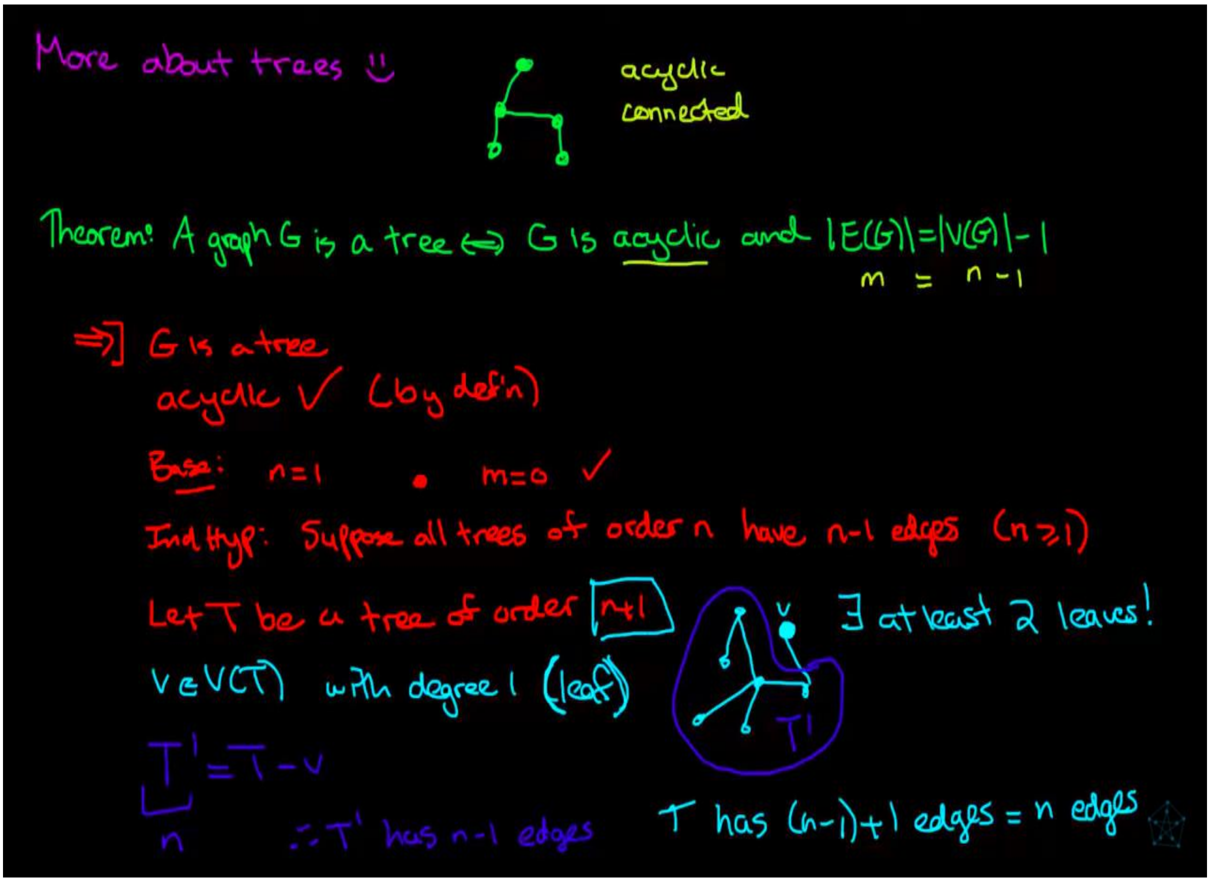
$S = (\cancel{7}, \cancel{6}, \cancel{6}, \cancel{1})$

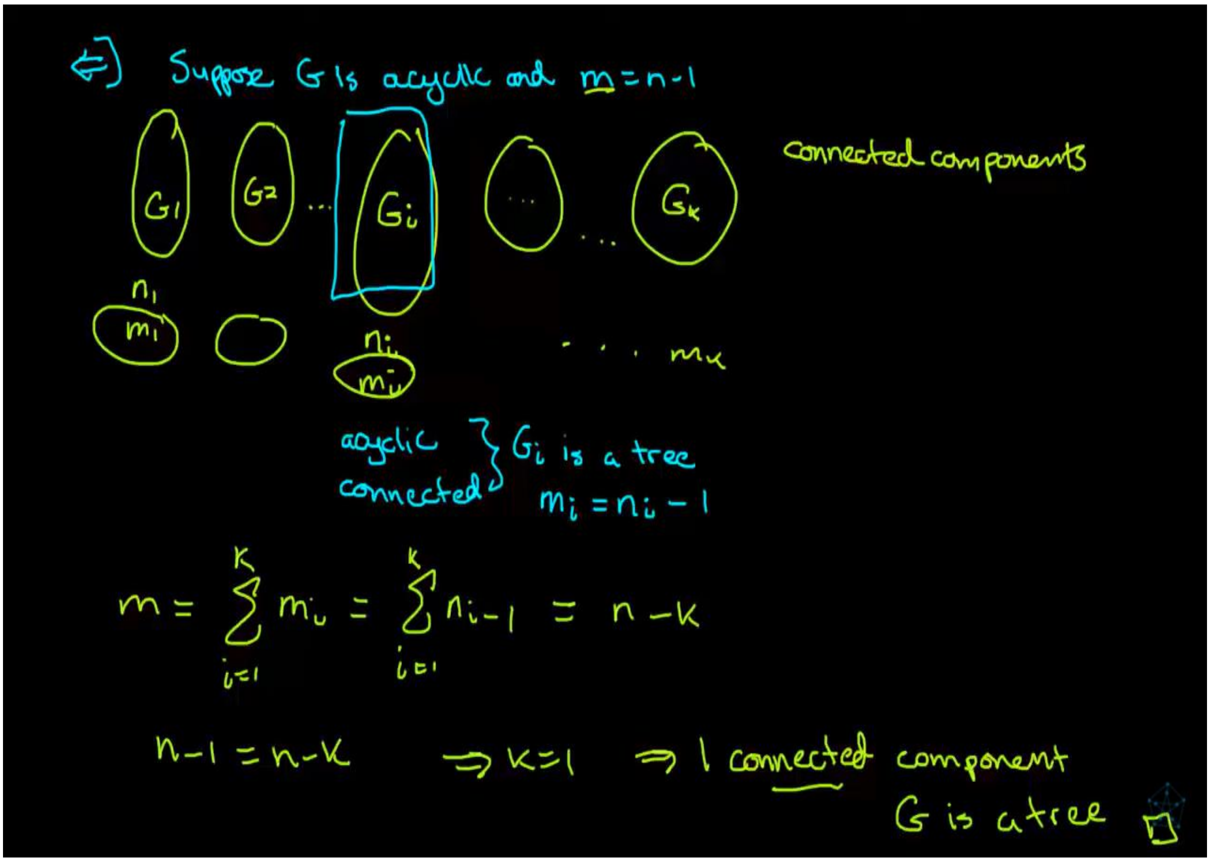
$L = \{1, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6}, 7\}$


delete y from L
• Continue until 2 items remain in L
↳ join these two via an edge
S = (4, 7, 6, 6, 1) length 5 ⇒ n = 7
L = {1, 2, ..., 7}
x ∈ L \ S smallest → x = 2
S = (7, 6, 6, 1)
L = {1, 3, 4, 5, 6, 7}
T
1
2
4
6
5
7
3

.

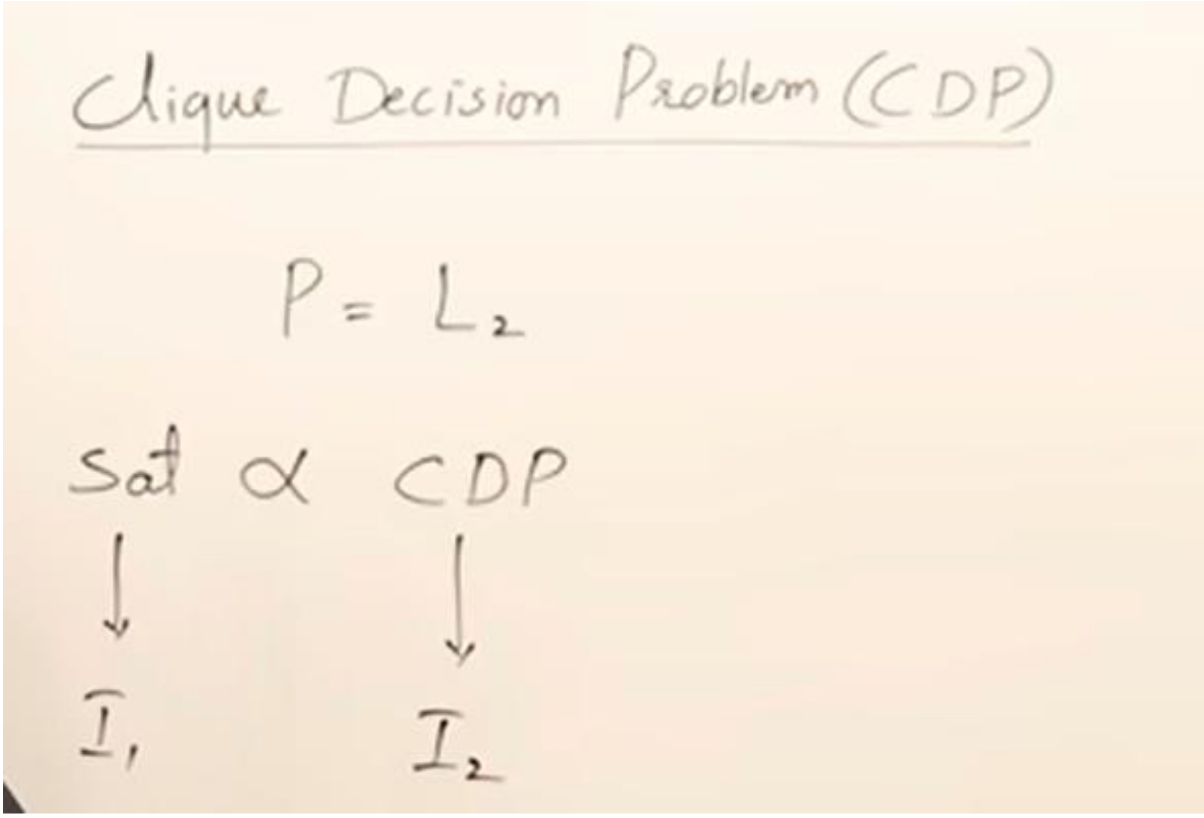
Q 3.a

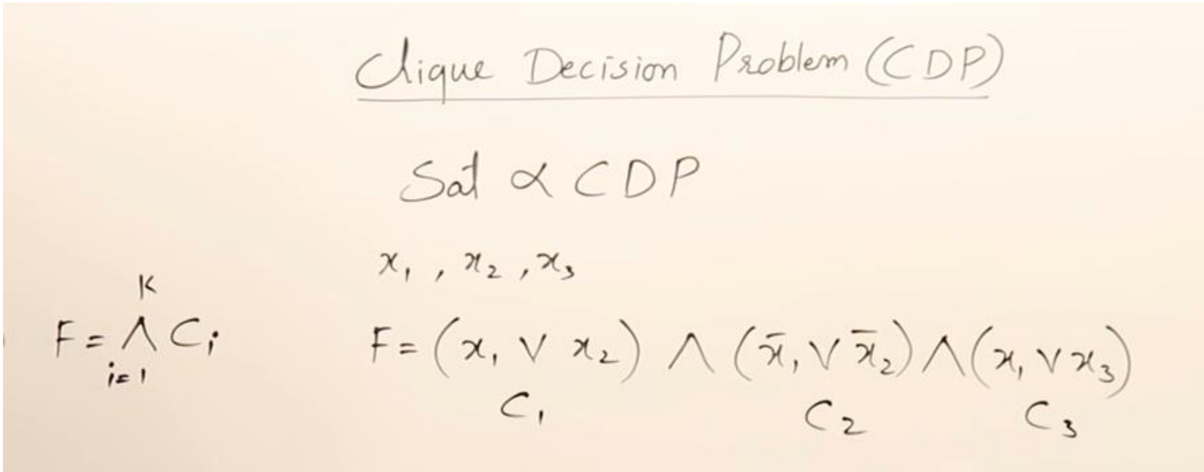

More about trees :)
acyclic
connected
Theorem: A graph G is a tree ⟺ G is acyclic and |E(G)| = |V(G)| - 1
m = n - 1
⇒] G is a tree
acyclic ✓ (by def'n)
Base: n=1 m=0 ✓
Ind Hyp: Suppose all trees of order n have n-1 edges (n ≥ 1)
Let T be a tree of order n+1
∃ at least 2 leaves!
v
T'
v ∈ V(T) with degree 1 (leaf)
T' = T - v
n
∴ T' has n-1 edges
T has (n-1)+1 edges = n edges



Xxxxxxxxxxxxxxxxxxxxx

Q3 b.


Clique Decision Problem (CDP)
$P = L_2$
Sat $\propto$ CDP
I_1
I_2


Clique Decision Problem (CDP)
Sat $\propto$ CDP
x_1, x_2, x_3
$F = \bigwedge_{i=1}^{K} C_i$
$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$
C_1
C_2
C_3

Sat $\propto$ CDP

x_1, x_2, x_3

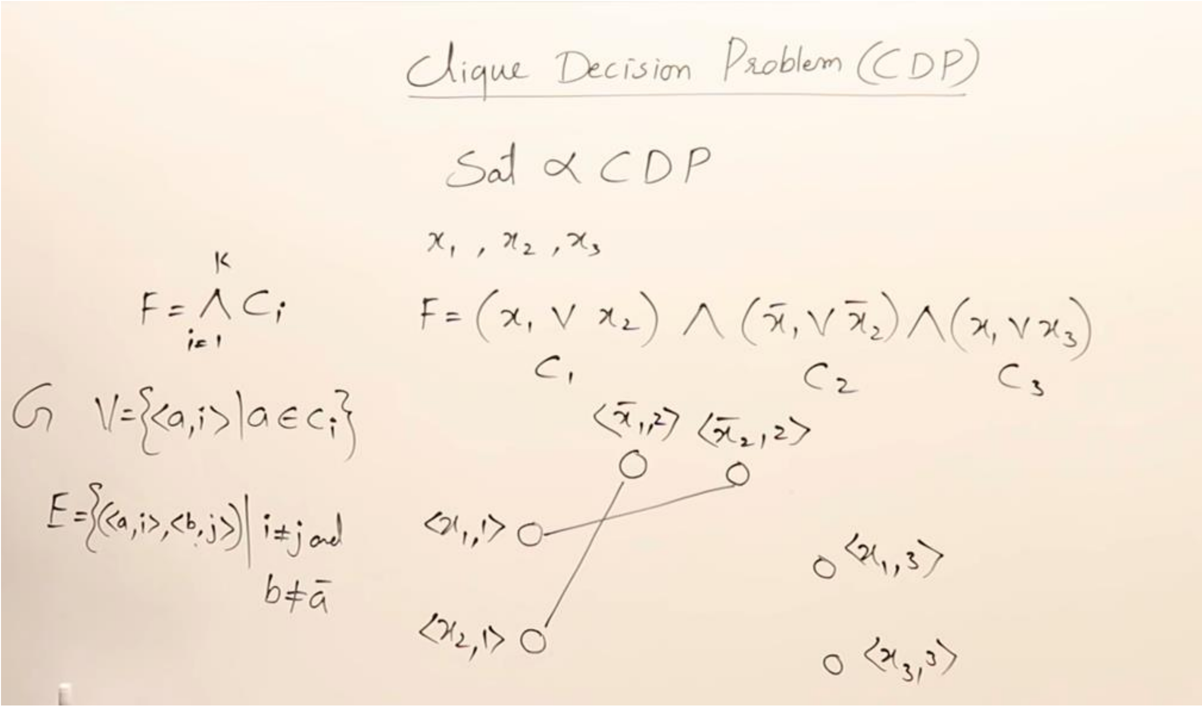
$$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$$

$C_1 \qquad C_2 \qquad C_3$

$\langle \bar{x}_1, 2 \rangle \quad \langle \bar{x}_2, 2 \rangle$

$\langle x_1, 1 \rangle \qquad\qquad \langle x_1, 3 \rangle$

$\langle x_2, 1 \rangle \qquad\qquad \langle x_3, 3 \rangle$

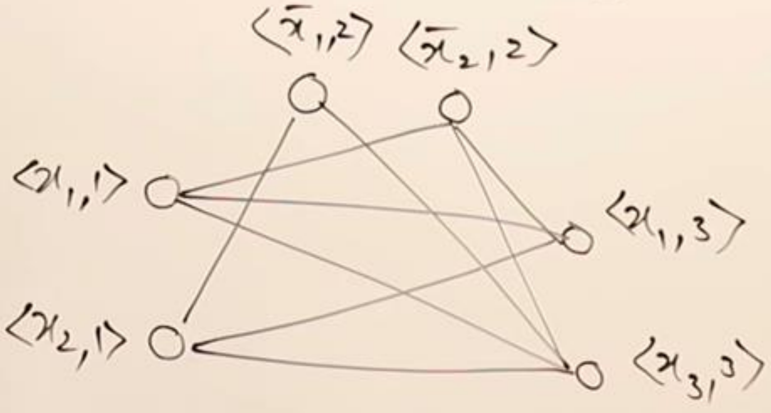

Clique Decision Problem (CDP)
Sat $\propto$ CDP
x_1, x_2, x_3
$F = \bigwedge_{i=1}^{k} C_i$
$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$
C_1
C_2
C_3
G $V = \{\langle a,i\rangle \mid a \in C_i\}$
$E = \{(\langle a,i\rangle, \langle b,j\rangle) \mid i \neq j$ and $b \neq \bar{a}\}$
$\langle \bar{x}_1, 2\rangle$
$\langle \bar{x}_2, 2\rangle$
$\langle x_1, 1\rangle$
$\langle x_2, 1\rangle$
$\langle x_1, 3\rangle$
$\langle x_3, 3\rangle$

Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1 , x_2 , x_3

$$F = \underbrace{(x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{C_2} \wedge \underbrace{(x_1 \vee x_3)}_{C_3}$$

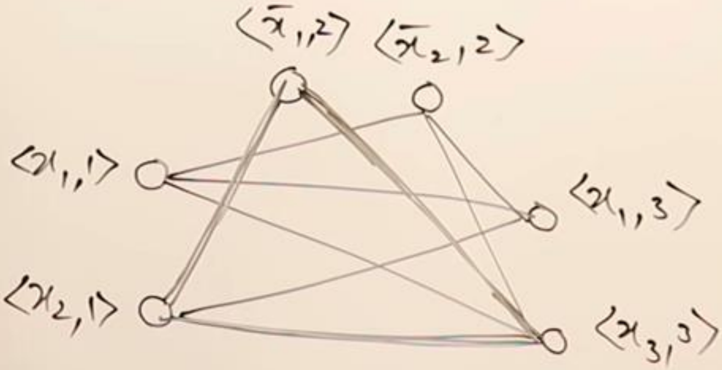


Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1, x_2, x_3 $\qquad$ $k=3$

$$F = \underbrace{(x_1 \lor x_2)}_{C_1} \land \underbrace{(\bar{x}_1 \lor \bar{x}_2)}_{C_2} \land \underbrace{(x_1 \lor x_3)}_{C_3}$$



$k=3$

Clique Decision Problem (CDP)

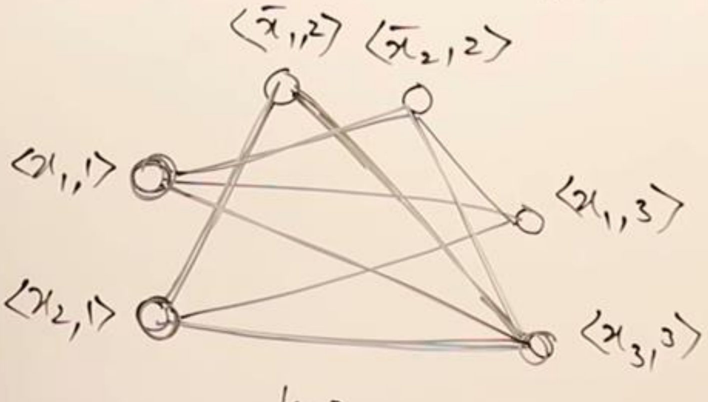
Sat $\propto$ CDP

x_1, x_2, x_3 $\qquad K=3$

$$F = (\overset{0}{x_1} \vee \overset{1}{x_2}) \wedge (\overset{1}{\bar{x}_1} \vee \overset{0}{\bar{x}_2}) \wedge (\overset{0}{x_1} \vee \overset{1}{x_3}) = 1$$

$C_1 = 1 \qquad C_2 = 1 \qquad C_3 = 1$

x_1	x_2	x_3
0	1	1



$k=3$

Q 4.a

Corollary 3.5. The genus g of a simple graph with n ($\geqslant 4$) vertices and $|E|$ edges satisfies:

$$g \geqslant \lceil \tfrac{1}{6}(|E|-3n)+1 \rceil$$

Proof. Every face of an embedding of the graph is bound by at least three edges each of which separates two faces, therefore $3f \leqslant 2.|E|$. From theorem 3.4, $g = \frac{1}{2}(|E|-n-f)+1)$ and so the result follows by substitution. ■

Specific results for thickness and genus are known for special cases (e.g., complete graphs, complete bipartite graphs (see exercise 3.11)) and involve lengthy proofs. In the case of complete graphs $|E| = \frac{1}{2}n(n-1)$ and the above corollaries then give:

$$g \geqslant \lceil \tfrac{1}{12}(n-3)\,(n-4) \rceil$$

and

$$T \geqslant \left\lceil \frac{n(n-1)}{6(n-2)} \right\rceil = \left\lfloor \frac{n(n-1)+(6n-14)}{6(n-2)} \right\rfloor = \lfloor \tfrac{1}{6}(n+7) \rfloor$$

It is known that in the result for g equality holds. Similarly, equality holds in the expression for T except for $n = 9$ and for $n = 10$, in both cases $T = 3$. These refinements required the considerable efforts of mathematicians over many years. Beineke & Wilson[7] provides a reference list of primary sources.

Filotti *et al.*[18] have described an $O(n^{O(g)})$-algorithm which takes as input a graph G and a positive integer g and which then finds an embedding of G on a surface of genus g if such an embedding exists.

Xxxxxxxxxxxxxxxxxxxxxxxx

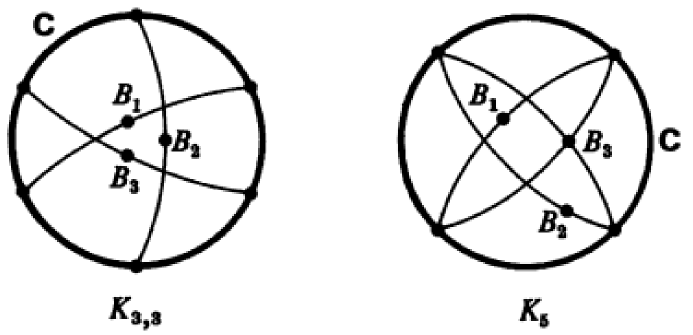
Q4.b

Theorem 3.6. A necessary and sufficient condition for a graph G to be planar is that for every circuit C of G the auxiliary graph $G^{+}(C)$ is bipartite.

Proof. The condition is necessary because for any circuit C of a planar graph G, we can form a bipartition $(B, \bar{B})$ of the bridge vertices of G relative to C, such that bridges in B lie in one face of C for $\tilde{G}$, and the bridges of $\bar{B}$ lie in the other face. Clearly, $G^{+}(C)$ is bipartite because no edge of $G^{+}(C)$ connects two vertices in B or connects two vertices in $\bar{B}$.

That the condition is sufficient can be seen as follows. If G is not planar then according to Kuratowski's theorem G contains a subgraph homeomorphic to K_5 or to $K_{3,3}$. We suppose that G contains K_5 or $K_{3,3}$ as a subgraph, the generalisation to G containing proper homeomorphisms is obvious. In either case (see figure 3.12, in which the chosen circuits are

Fig. 3.12



indicated by heavily scored edges), we can choose C of the subgraph such that $G^{+}(C)$ is not bipartite. For $K_{3,3}$ there are three bridges B_1, B_2 and B_3, each of which is a single edge and any two of which are incompatible. In the case of K_5 there are again three bridges B_1, B_2 and B_3. B_1 and B_2 are single edges while B_3 is a vertex of K_5 plus its edges of attachment to C. Again any two of the bridges are incompatible. Thus for both K_5 and $K_{3,3}$, for the circuits chosen, $G^{+}(C) = K_3$ which is not bipartite. ∎

Q8. b

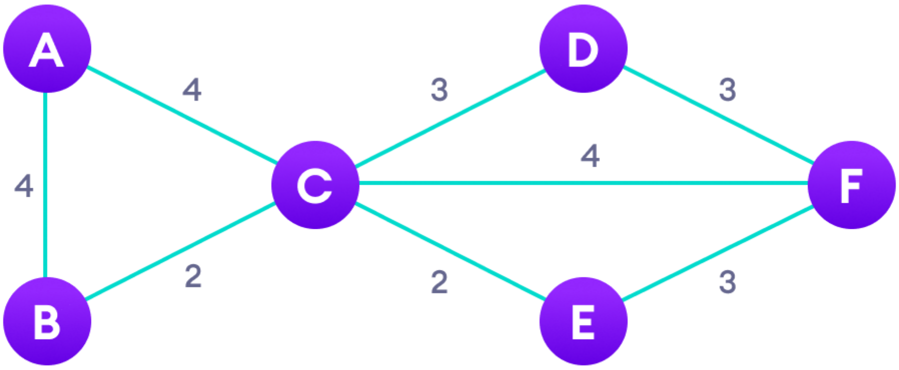
How Prim's algorithm works

It falls under a class of algorithms called greedy algorithms that find the local optimum in the hopes of finding a global optimum.
We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

The steps for implementing Prim's algorithm are as follows:

1. Initialize the minimum spanning tree with a vertex chosen at random.
2. Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree
3. Keep repeating step 2 until we get a minimum spanning tree

Example of Prim's algorithm

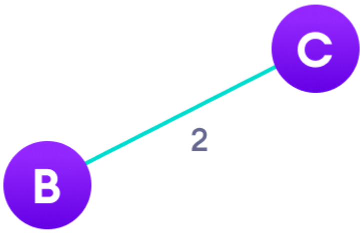


Step: 1

Start with a weighted graph

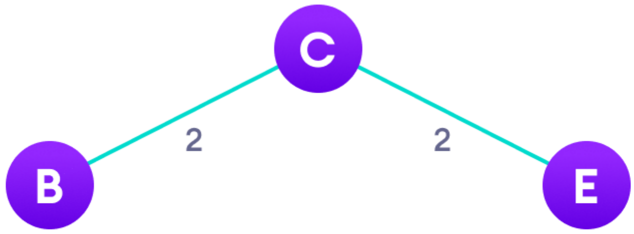
Step: 2

Choose a vertex



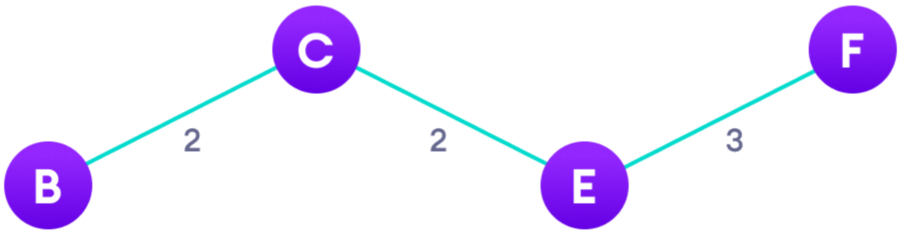
Step: 3

Choose the shortest edge from this vertex and add it



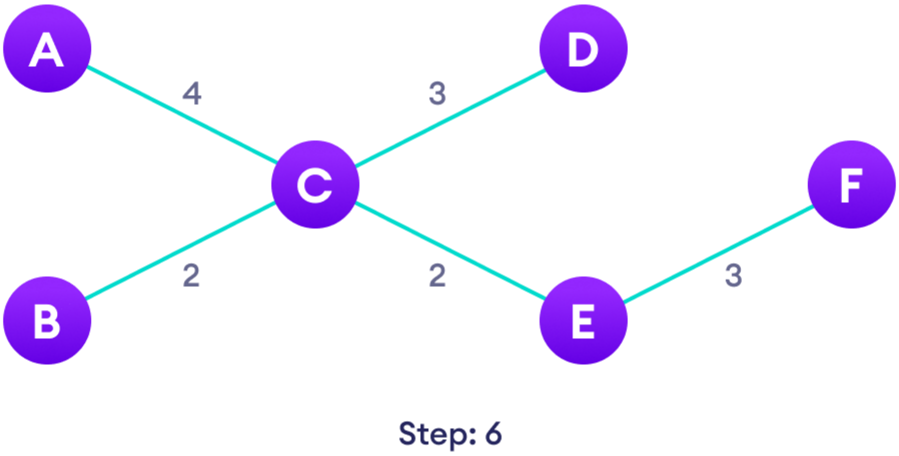
Step: 4

Choose the nearest vertex not yet in the solution



Step: 5

Choose the nearest edge not yet in the solution, if there are multiple choices, choose one at random



Repeat until you have a spanning tree

Xxxxxxxxxxxxxxxxx

Q 5 a.

Graph Coloring and Schedules

EG: Suppose want to schedule some final exams for CS courses with following call numbers:

1007, 3137, 3157, 3203, 3261, 4115, 4118, 4156

Suppose also that there are no common students in the following pairs of courses because of prerequisites:

1007-3137
1007-3157, 3137-3157
1007-3203
1007-3261, 3137-3261, 3203-3261
1007-4115, 3137-4115, 3203-4115, 3261-4115
1007-4118, 3137-4118
1007-4156, 3137-4156, 3157-4156

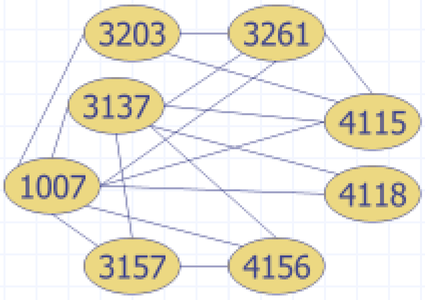
How many exam slots are necessary to schedule exams?

Graph Coloring and Schedules

Turn this into a graph coloring problem. Vertices are courses, and edges are courses which *cannot* be scheduled simultaneously because of possible students in common:

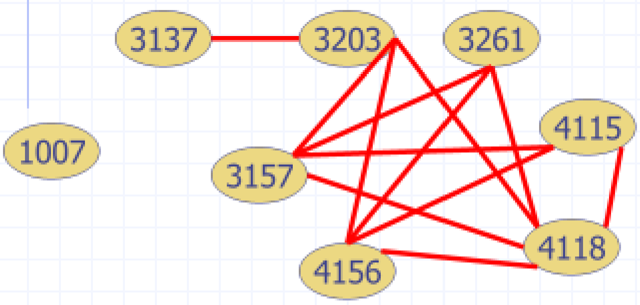
Graph Coloring and Schedules

One way to do this is to put edges down where students mutually excluded…



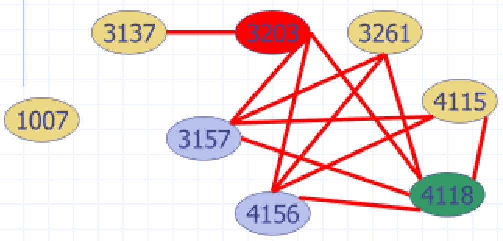
Graph Coloring and Schedules

Is 3-colorable. Try to color by Red, Green, Blue.



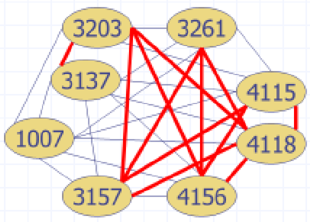
Graph Coloring and Schedules

So 4156 must be Blue:



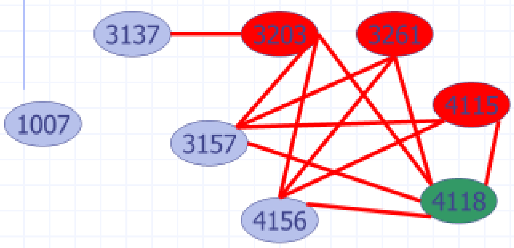
Graph Coloring and Schedules

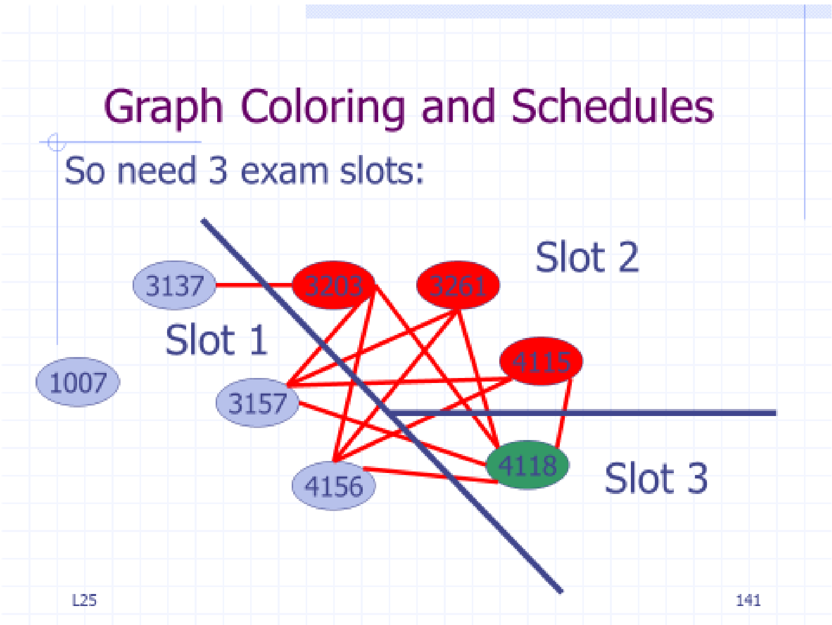
...and then compute the complementary graph:



Graph Coloring and Schedules

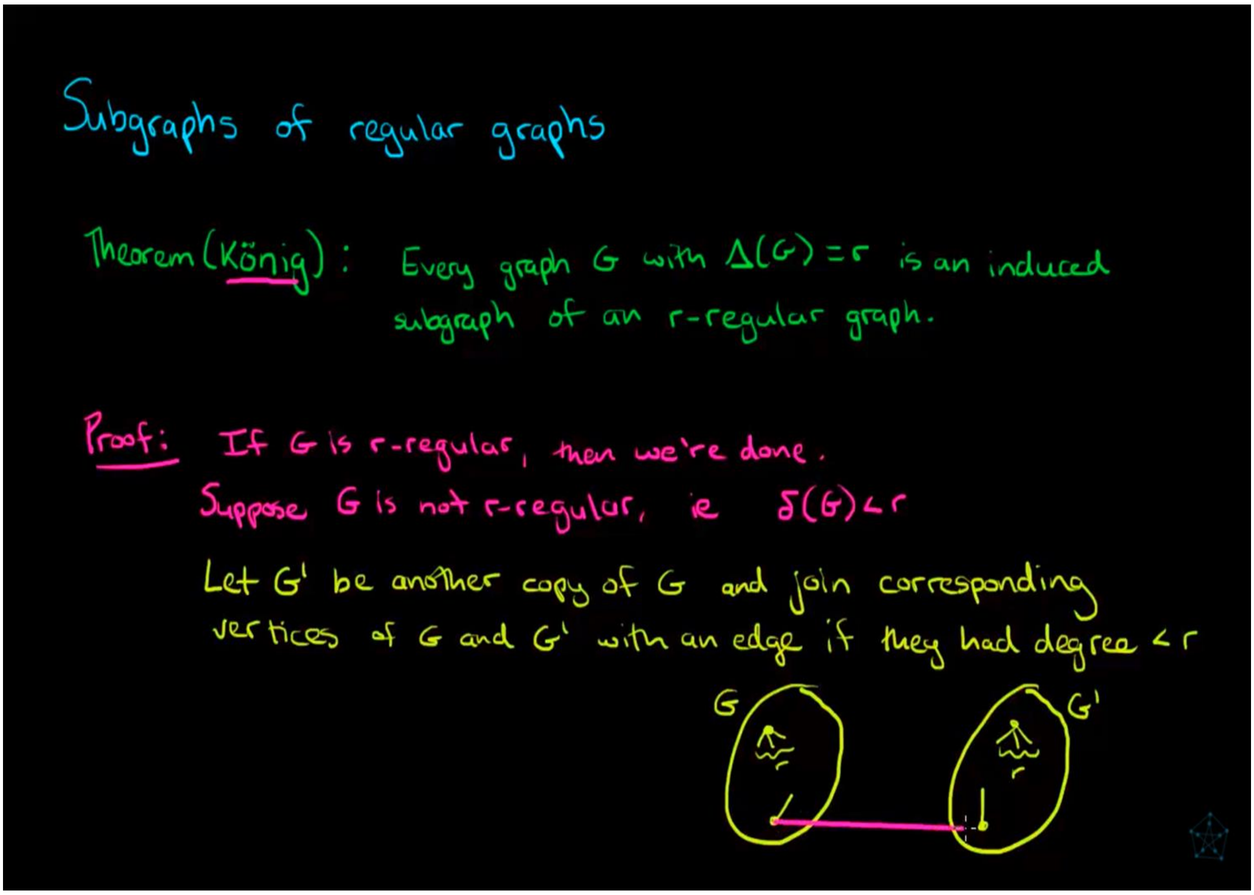
3137 and 1007 easy to color.





Xxxxxxxxxxxxxxxxx

Ans. 5.b

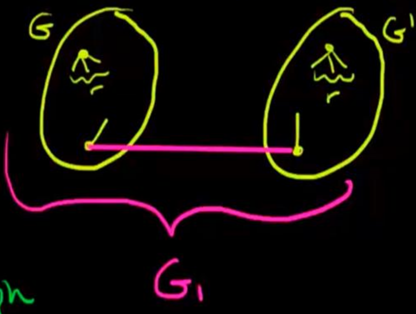

Subgraphs of regular graphs
Theorem (König): Every graph G with Δ(G) = r is an induced subgraph of an r-regular graph.
Proof: If G is r-regular, then we're done.
Suppose G is not r-regular, ie δ(G) < r
Let G' be another copy of G and join corresponding vertices of G and G' with an edge if they had degree < r
G
G'

Proof: If G is r-regular, then we're done.

Suppose G is not r-regular, ie $\delta(G) < r$

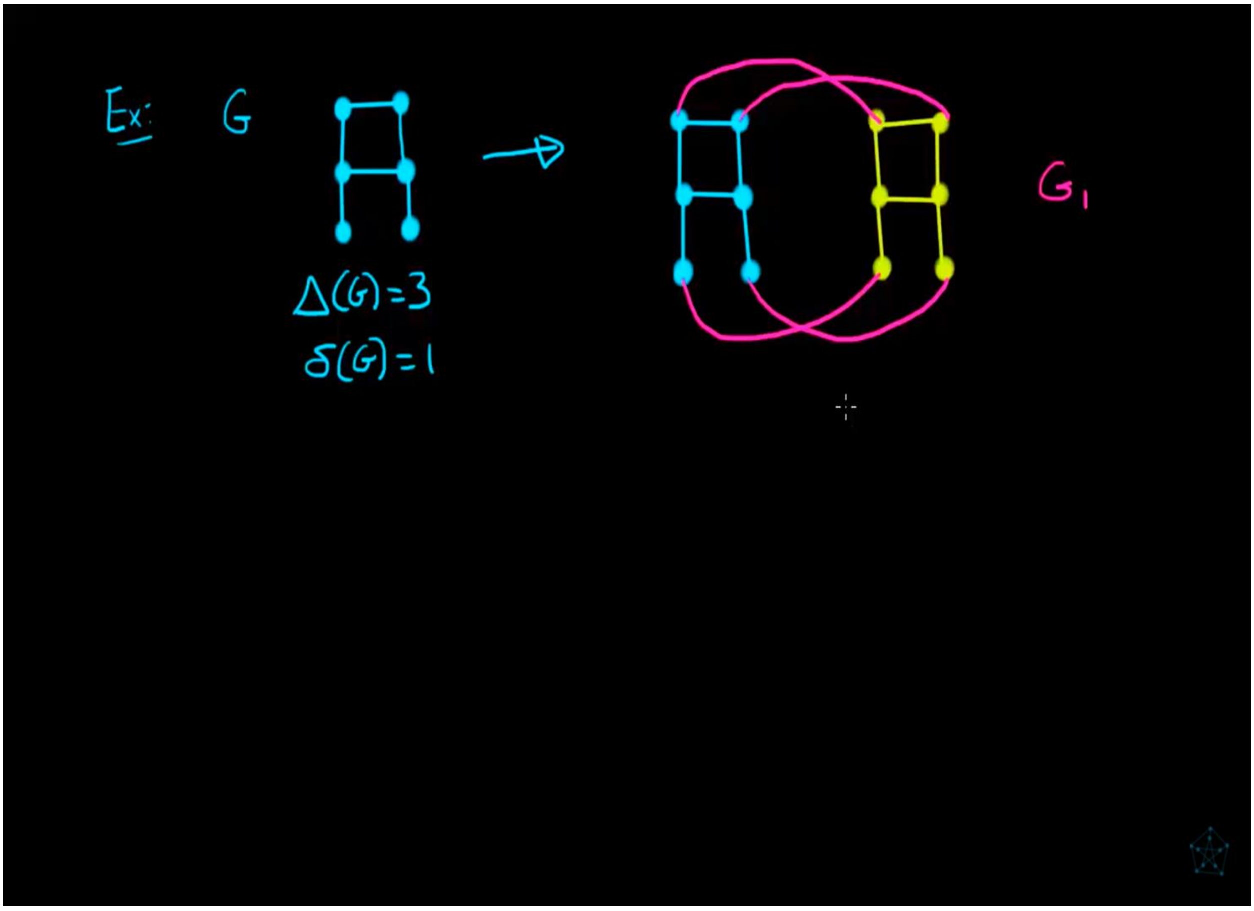
Let G' be another copy of G and join corresponding vertices of G and G' with an edge if they had degree $< r$

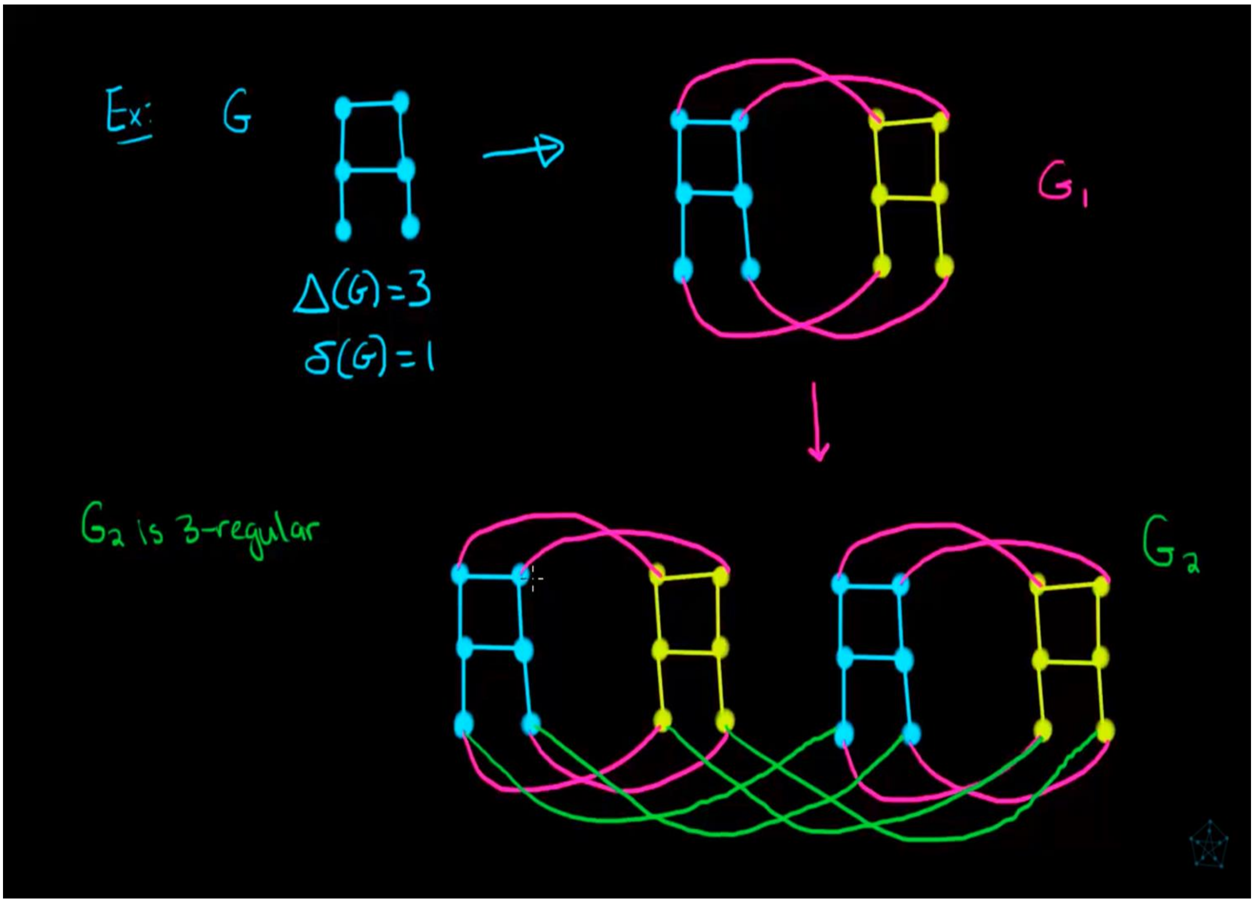
Call the resulting graph G_1

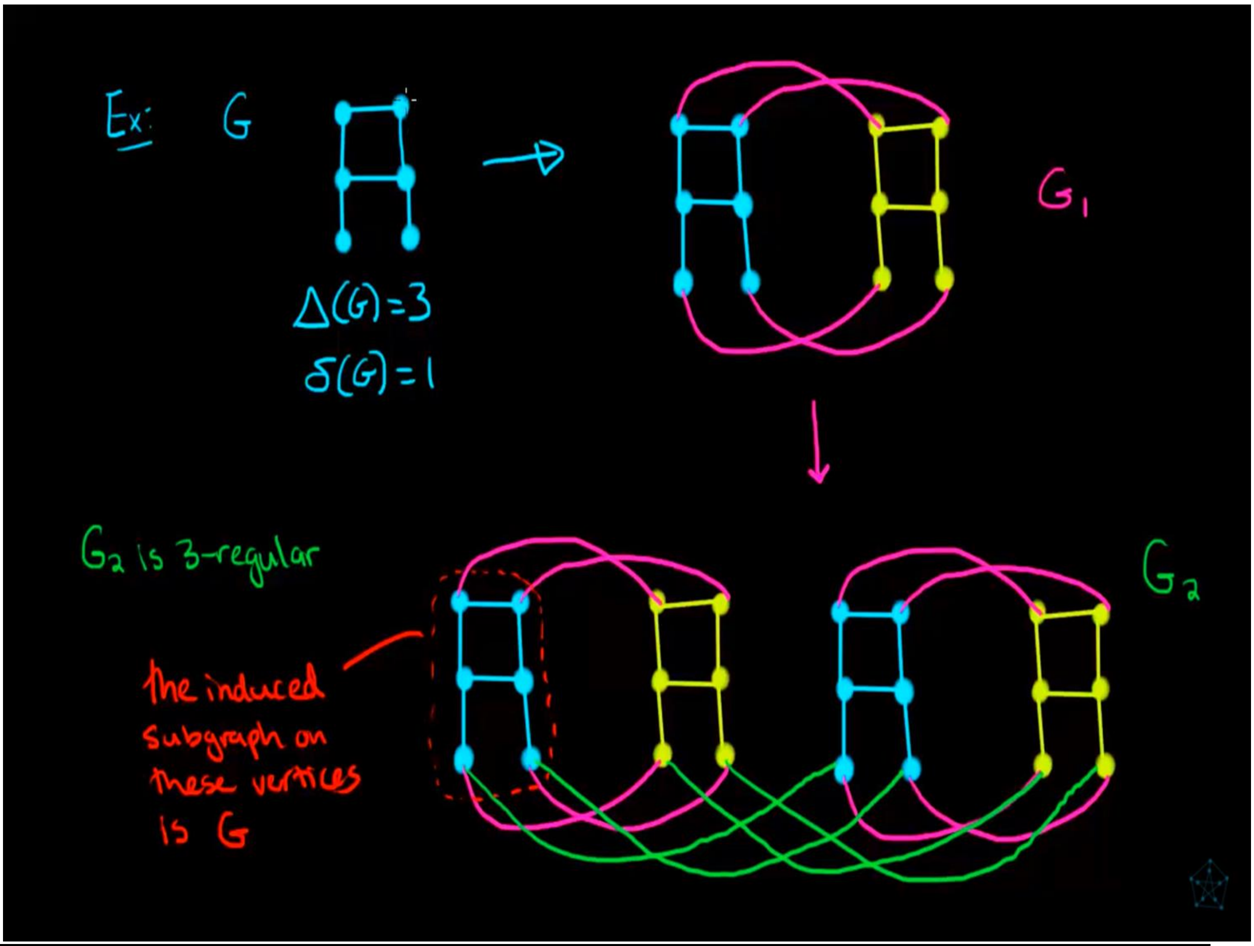


If G_1 is r-regular, then we're done since G is an induced subgraph of G_1.

If not, then continue the procedure until arriving at an r-regular graph G_k where $k = r - \delta(G)$


Ex: G
Δ(G)=3
δ(G)=1
G₁


Ex: G
Δ(G)=3
δ(G)=1
G1
G2 is 3-regular
G2



----------------xx-------------------------

Ans. 6 a. If a graph G is maximal planar graph with *n* vertices (*n*>=*3*) and *m* edges then show that *m* = *3n* – *6*.

Answer:

Theorem: If G is a maximal planar graph with n vertices $(n \geq 3)$ and m edges then $\boxed{m = 3n - 6}$

Proof:

Take a plane drawing of G with r regions. The boundary of every region is a triangle.

$$\underbrace{\sum_{\text{all regions } R} (\#\text{edges on boundary } R)}_{= 2m} = 3r$$

each edge is on the boundary of 2 regions

Euler's Formula $\Rightarrow$ $n - m + r = 2$

$$n - m + \frac{2m}{3} = 2$$

$$3n - 3m + 2m = 6 \Rightarrow 3n - 6 = m$$

Euler's Formula $\Rightarrow$ $n - m + r = 2$

$$n - m + \frac{2m}{3} = 2$$

$$3n - 3m + 2m = 6 \Rightarrow 3n - 6 = m$$

■

Corollary: If G is a planar graph with $n \geq 3$ vertices and m edges then
$$m \leq 3n - 6$$

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

ANSWER: 6. b

Theorem: If G is a connected plane graph with n vertices, m edges & r regions, then

$$\boxed{n-m+r=2}$$ (Euler's Formula)

Proof (induction on m)

Basis: $m=0$

Then $G=K_1$ so $\left.\begin{matrix} n=1 \\ m=0 \\ r=1 \end{matrix}\right\} n-m+r=1-0+1=2$ ✓

Ind. Hyp: Suppose the theorem is true for all connected plane graphs with $< m$ edges (where $m \geq 1$)

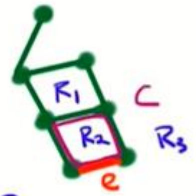
Now consider a connected plane graph G on m edges, n vertices, r regions

Ind. Hyp: Suppose the theorem is true for all connected plane graphs with $< m$ edges (where $m \geq 1$)

Now consider a connected plane graph G on m edges, n vertices, r regions

- If G is a tree
  then $m = n-1$ and $r = 1$
  so $n - m + r = n - (n-1) + 1 = 2$ ✓

- If G is not a tree then G has a cycle C
  Let e be an edge of C
  Then e is not a bridge
  $\therefore$ $G \setminus e$ is connected and planar
  and has n vertices
  $m-1$ edges ←
  $r-1$ regions



By the Ind Hyp. the theorem holds for $G \setminus e$

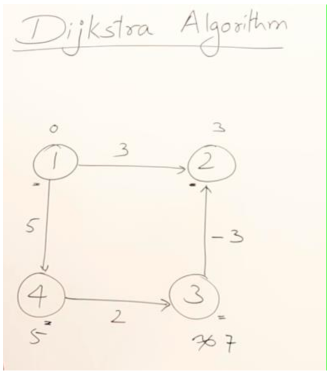
$\therefore$ $n - (m-1) + (r-1) = 2$

$\therefore$ $n - m + r = 2$ ✓

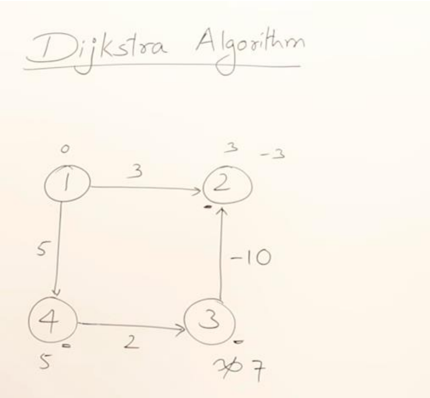
---------------xx------------------------

Q 8 a.

Successful D.A.

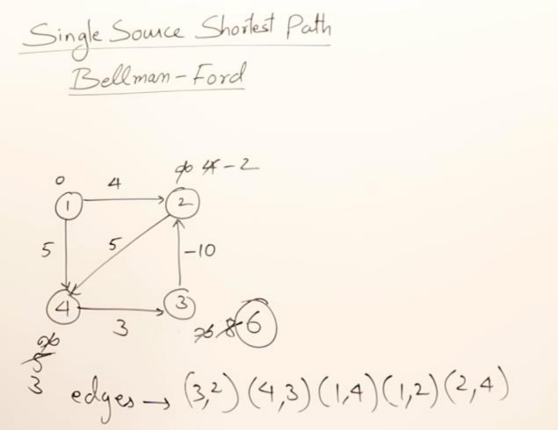


Un Successful D.A.



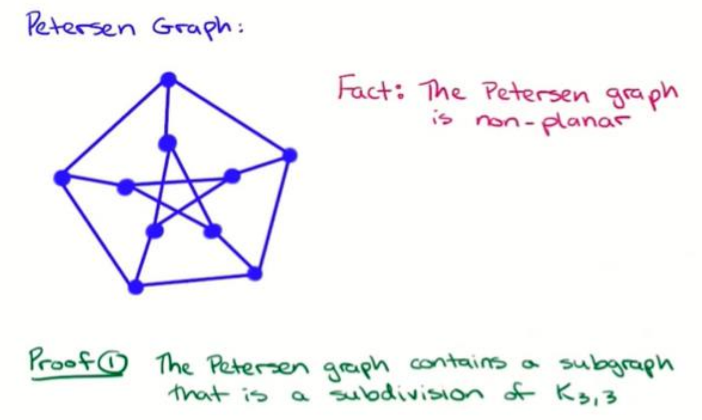
For Negative weights

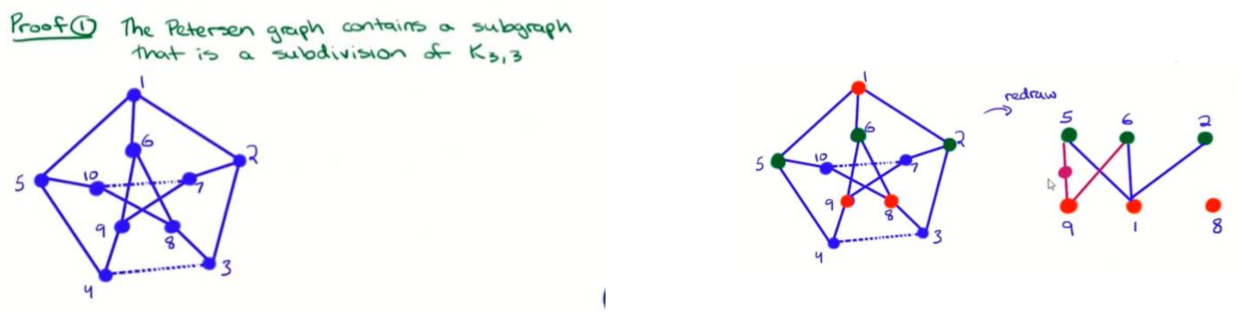
Situation for which Bellman ford will fail

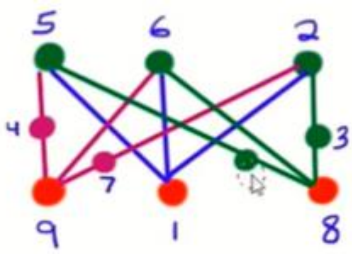


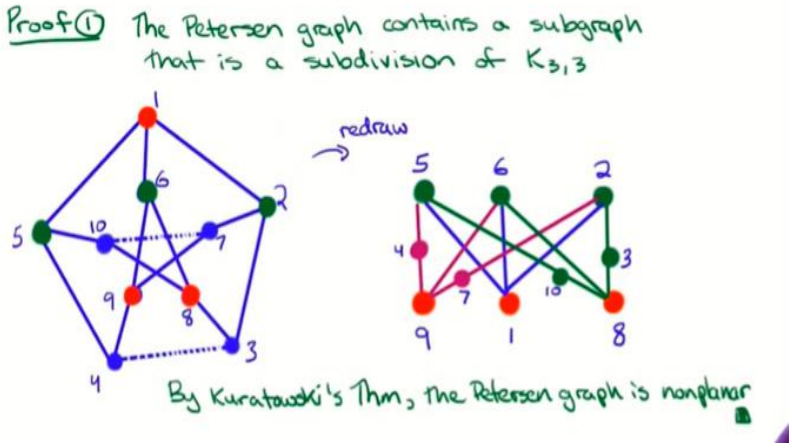
----------------xx-------------------------

Ans. 9. a & b


Petersen Graph:
Fact: The Petersen graph is non-planar
Proof ① The Petersen graph contains a subgraph that is a subdivision of $K_{3,3}$

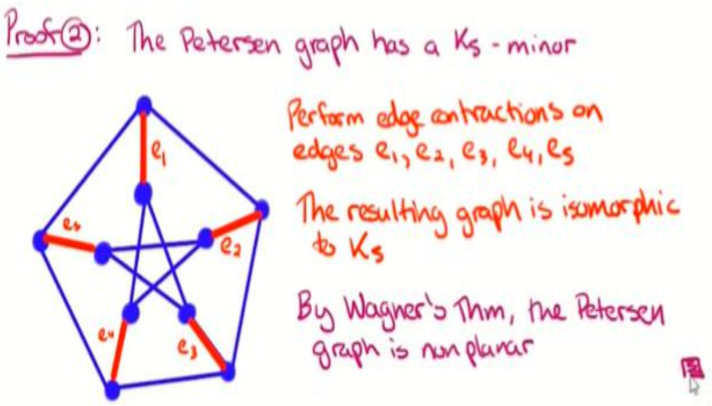

Proof ① The Petersen graph contains a subgraph that is a subdivision of $K_{3,3}$
1
6
2
5
10
7
9
8
4
3
redraw
5
6
2
9
1
8


5
6
2
4
3
7
9
1
8



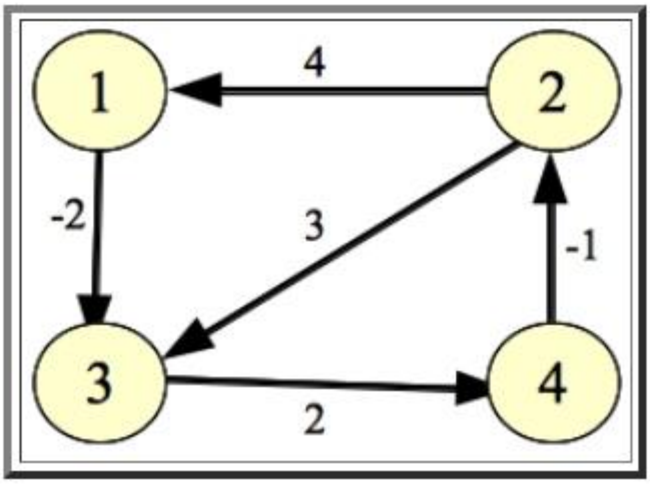
Prove that Peterson Graph is non-planar using Wagner's Theorem.

Mention def. of Minor

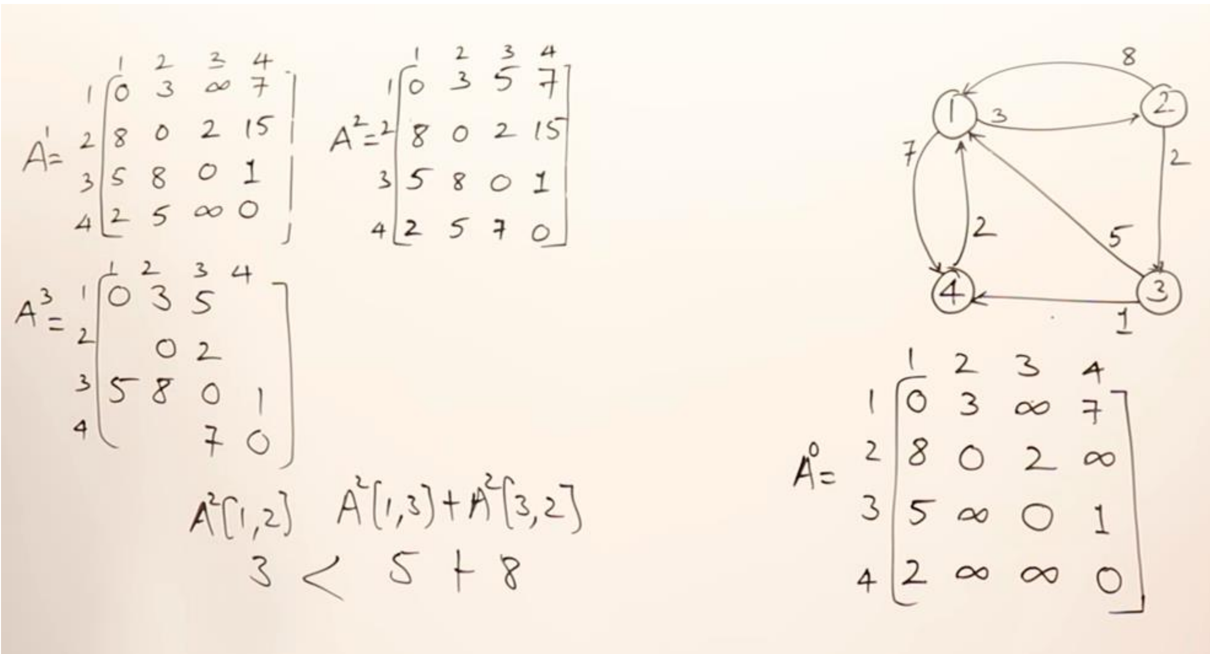


Then draw the resulting graph.

9 a. The Floyd-Warshall Algorithm example



----------------xx-------------------------



Q 7. a open question to answer

Q 7. b Prove that every planar graph has a dual.

Answer: Solved as theorem in Alan Gibson Book

----------------xx-------------------------

Q. 10 b . Describe a method for finding a minimum connected dominating set and also illustrate the same with suitable example.

Answer:

Bipartite Graph & Set Covering Problem

- The Connected Dominating Set (CDS) problem is to find a minimum connected dominating set.
- Finding a minimum connected dominating set is known to be an NP-complete problem.
- This essentially means that these class of problems cannot be solved quickly (in polynomial time).
- Several authors have proposed algorithms for obtaining approximate minimal connected dominating sets.

Connected Dominating Sets

- The problem of finding a minimum connected dominating set can be mapped into a Set Covering Problem.
- The Set Covering Problem is essentially a problem concerning bipartite graphs that can be stated as follows.

Bipartite Graph & Set Covering Problem

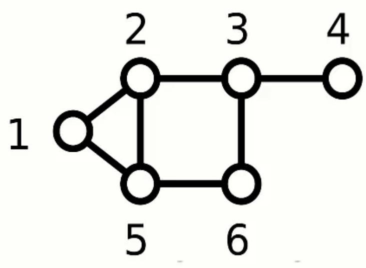
- Suppose that H is a bipartite graph, consisting of two sets of nodes A and B, where edges only make connections between A and B.
- Also assume that for each node in *B*, there is at least one edge connecting it to a node in *A*.
- The goal is to find a minimal subset C of A such that every node in B is covered by (i.e., adjacent to) some node in C.

Bipartite Graph Vertex Covering Problem

- The bipartite graph vertex covering problem can be addressed using a greedy algorithm, which will be explained in the next lecture.
- At each stage, a vertex from A is elected that covers as many vertices from B as possible that are not yet covered by an elected vertex.
- Let us look at the following example.

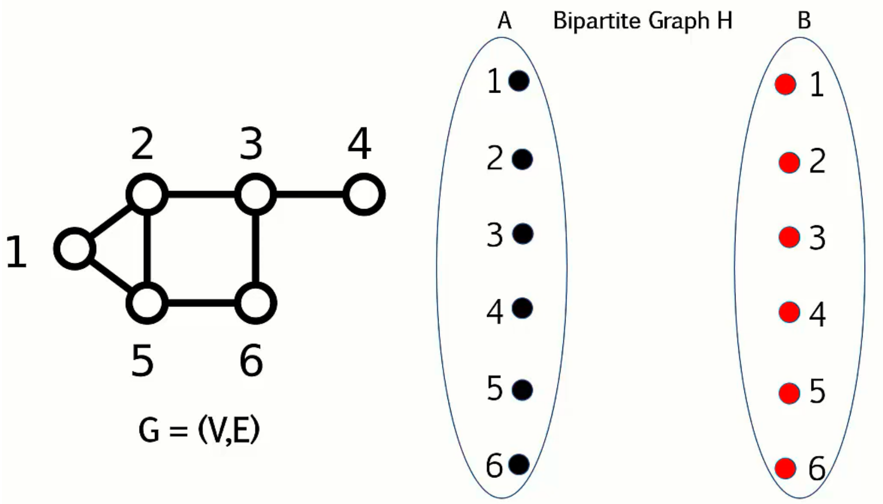
Bipartite Graph Example

- Let G be a connected graph (V,E).
- Let A and B are copies of vertices of E.
- Construct a bipartite graph H by putting an edge between vertices v of A and u of B if they are adjacent to each other.

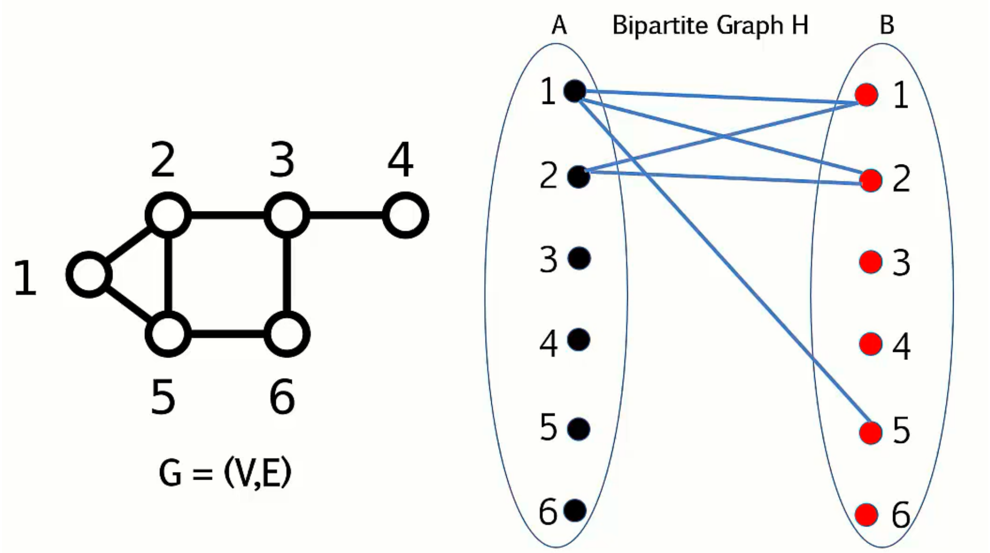


G = (V,E)

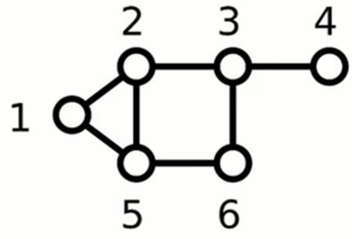
Bipartite Graph Example (Contd.)



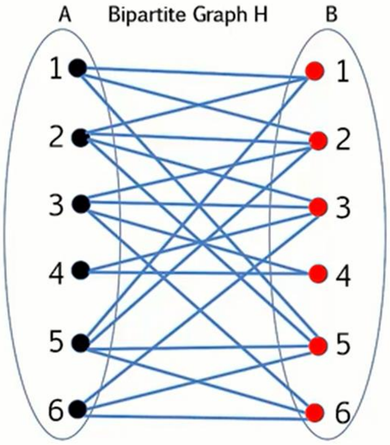
Bipartite Graph Example (Contd.)



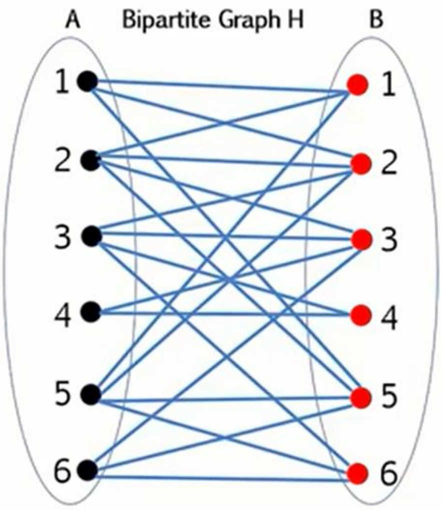
Bipartite Graph Example (Contd.)



G = (V,E)



Greedy Algorithm Example (Contd.)

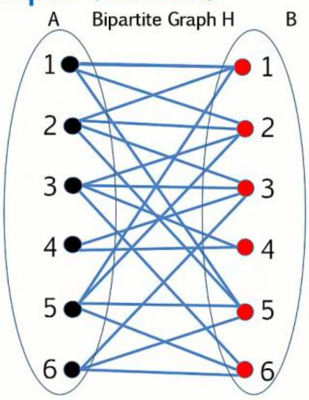


Compute the degree of all nodes in the set A in the bipartite graph H, i.e. compute the *covering numbers*.

Covering no. of vertex 1 (in A) = 3
Covering no. of vertex 2 (in A) = 4
Covering no. of vertex 3 (in A) = 4
Covering no. of vertex 4 (in A) = 2
Covering no. of vertex 5 (in A) = 4
Covering no. of vertex 6 (in A) = 3

Greedy Algorithm Example (Contd.)

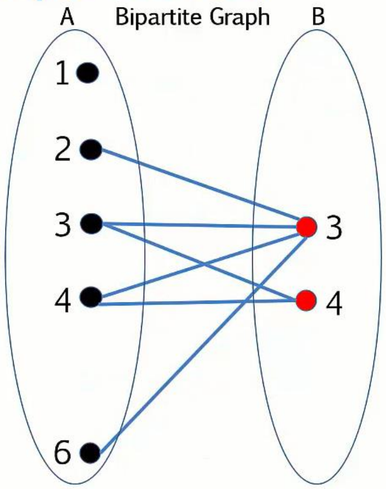
- Elect a node v from set A such that v has the highest covering number and add it to the output set C.
- If there is a tie, then the highest ID is used to break the tie.
- Remove all vertices in the set B that are covered by the node v.
- Also remove v from A.
- After the first round, C = {5} as vertex 5 from A has the highest covering no. as well as the highest ID.



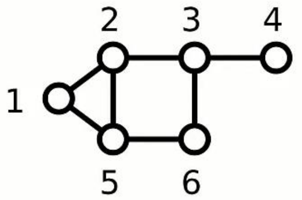
Greedy Algorithm Example (Contd.)

- During the second round, re-compute the covering no. of all remaining vertices in A.
- Although vertices 3 & 4 in A have the same covering no., vertex 4 is selected because it has the highest id.
- After the second round, all vertices in B are exhausted and the algorithm converges.
- The final output set C = {4,5} is a minimal dominating set.

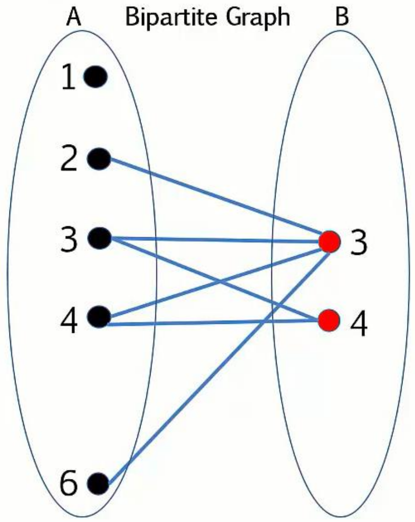
Covering no. of vertex 1 (in A) = 0
Covering no. of vertex 2 (in A) = 1
Covering no. of vertex 3 (in A) = 2
Covering no. of vertex 4 (in A) = 2
Covering no. of vertex 6 (in A) = 1



Greedy Algorithm For MCDS

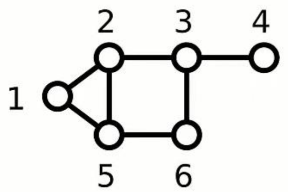


G = (V,E)

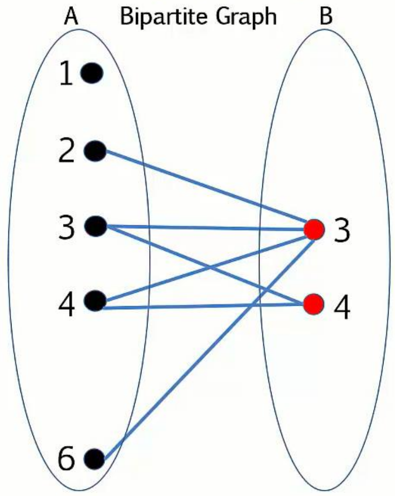


- We can adapt the same greedy algorithm to find a minimal connected dominating set.
- It will be helpful to identify the vertices of A with vertices of G.
- After we select vertex 5 from A, only a vertex adjacent to 5 in G can be selected next.

Greedy Algorithm For MCDS (Contd.)



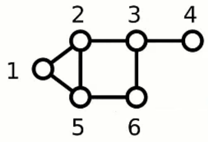
G = (V,E)



- 1, 2 and 6 are adjacent to 5 in G.
- 2 and 6 have the same covering no.
- We select 6 from A as it has the highest id.
- Then we remove vertex 6 from A and vertex 3 from B.
- Now the output set C = {5,6}

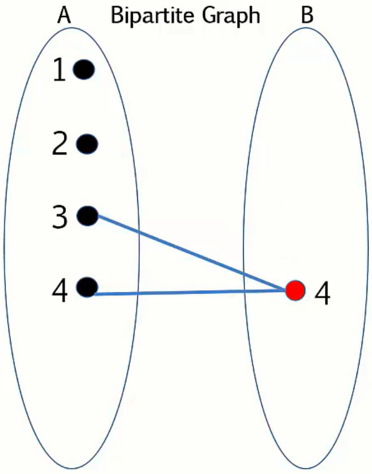
Covering no. of vertex 1 = 0
Covering no. of vertex 2 = 1
Covering no. of vertex 6 = 1

Greedy Algorithm For MCDS (Contd.)

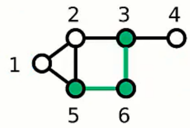


G = (V,E)

Covering no. of vertex 1 = 0
Covering no. of vertex 2 = 0
Covering no. of vertex 3 = 1



- We must choose a vertex in A with highest covering no.
- 4 is not eligible as it neither adjacent to 5 or 6 in G.
- The eligible vertices are 1, 2 and 3.
- We select 3 as it has the highest covering no.
- Now the output set C = {3,5,6} is a minimal connected dominating set.



----------------XX------------------------

Q 6 a. Answer:

Theorem: If G is a maximal planar graph with n vertices $(n \geq 3)$ and m edges then $\boxed{m = 3n - 6}$

Proof:

Take a plane drawing of G with r regions.
The boundary of every region is a triangle.

$$\underbrace{\sum_{\substack{\text{all regions} \\ R}} (\#\text{edges on boundary } R)}_{= 2m} = 3r$$

each edge is on the boundary of 2 regions

Euler's Formula $\Rightarrow$ $n - m + r = 2$

$$n - m + \frac{2m}{3} = 2$$

$$3n - 3m + 2m = 6 \quad \Rightarrow 3n - 6 = m$$

Euler's Formula $\Rightarrow$ $n - m + r = 2$

$$n - m + \frac{2m}{3} = 2$$

$$3n - 3m + 2m = 6 \quad \Rightarrow 3n - 6 = m$$

$\blacksquare$

Corollary: If G is a planar graph with $n \geq 3$ vertices and m edges then

$$m \leq 3n - 6$$

----------------xx-------------------------

6b. For a planar graph G with *n* number of vertices and *e* number of edges with *r* regions, prove by method of induction that $n - e + r = 2$. *Or* $N - E + f = 2$.

Q 6 b. ANSWER:

Theorem: If G is a connected plane graph with n vertices, m edges & r regions, then

$\boxed{n - m + r = 2}$ (Euler's Formula)

Proof (induction on m)

Basis: $m = 0$

Then $G = K_1$ so $\left.\begin{matrix} n = 1 \\ m = 0 \\ r = 1 \end{matrix}\right\} n - m + r = 1 - 0 + 1 = 2$ ✓

Ind. Hyp: Suppose the theorem is true for all connected plane graphs with $< m$ edges (where $m \geq 1$)

Now consider a connected plane graph G on m edges, n vertices, r regions

Ind. Hyp: Suppose the theorem is true for all connected plane graphs with $< m$ edges (where $m \geq 1$)

Now consider a connected plane graph G on m edges, n vertices, r regions

- If G is a tree

  then $m = n-1$ and $r = 1$

  so $n - m + r = n - (n-1) + 1 = 2$ ✓

- If G is not a tree then G has a cycle C

  Let e be an edge of C

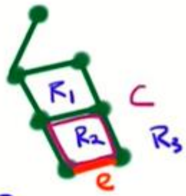
  Then e is not a bridge

  $\therefore G \setminus e$ is connected and planar

  and has n vertices

  $m-1$ edges ←

  $r-1$ regions



removing e means regions R_2 & R_3 join into 1 region

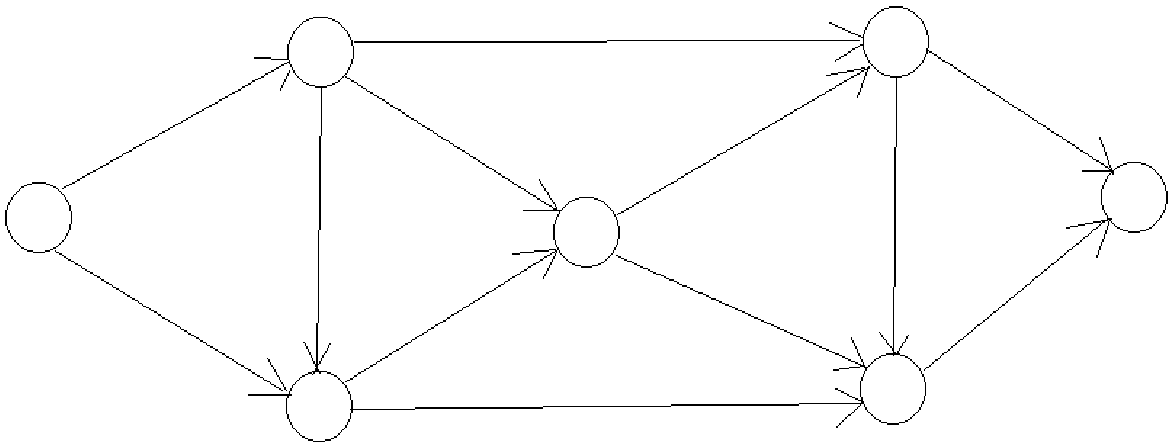
By the Ind Hyp. the theorem holds for $G \setminus e$

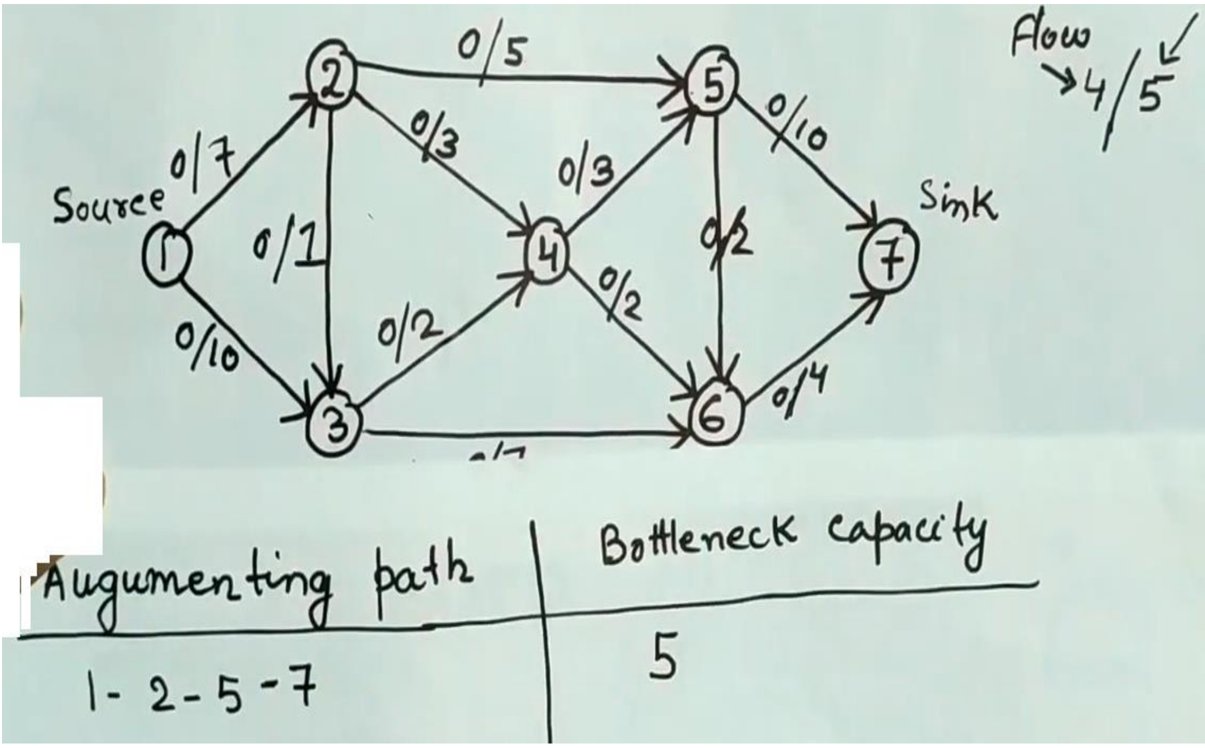
$$\therefore \quad n - (m-1) + (r-1) = 2$$

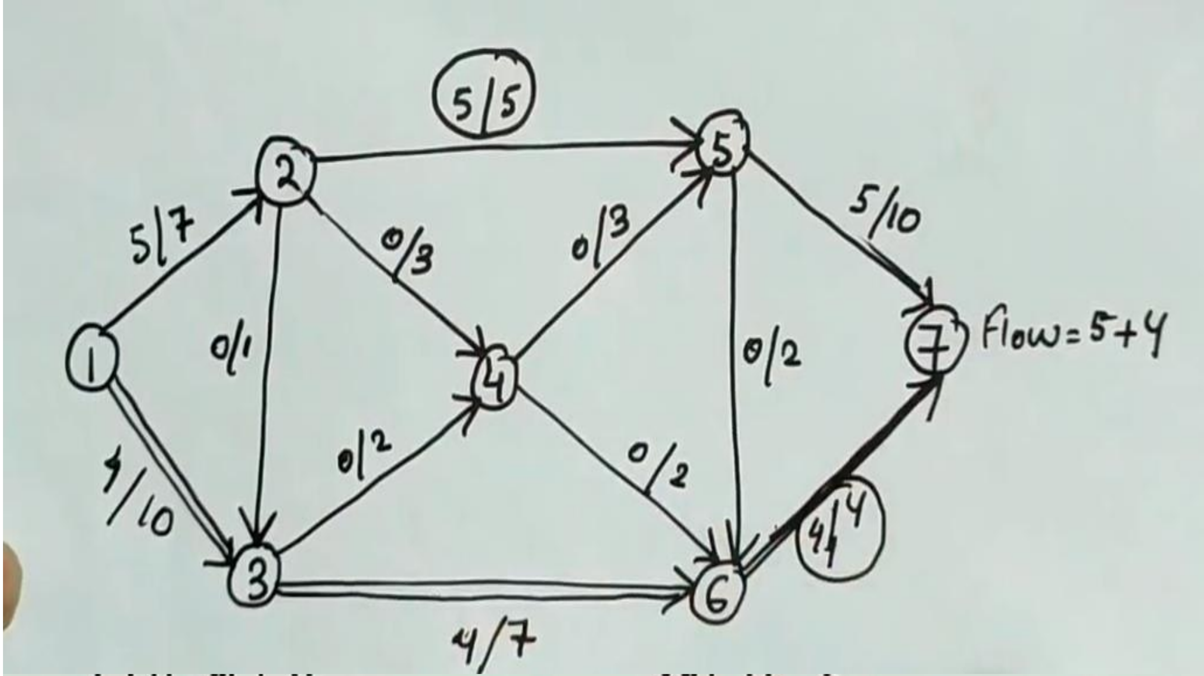
$$\therefore \quad n - m + r = 2 \quad ✓$$

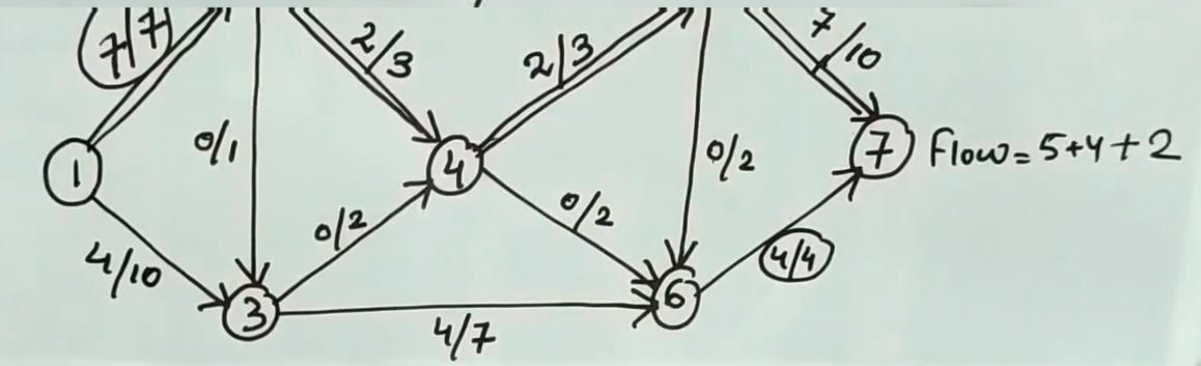
---------------xx-------------------------

Q10 a Find the maximum flow through the given network using Ford-Fulkerson algorithm.

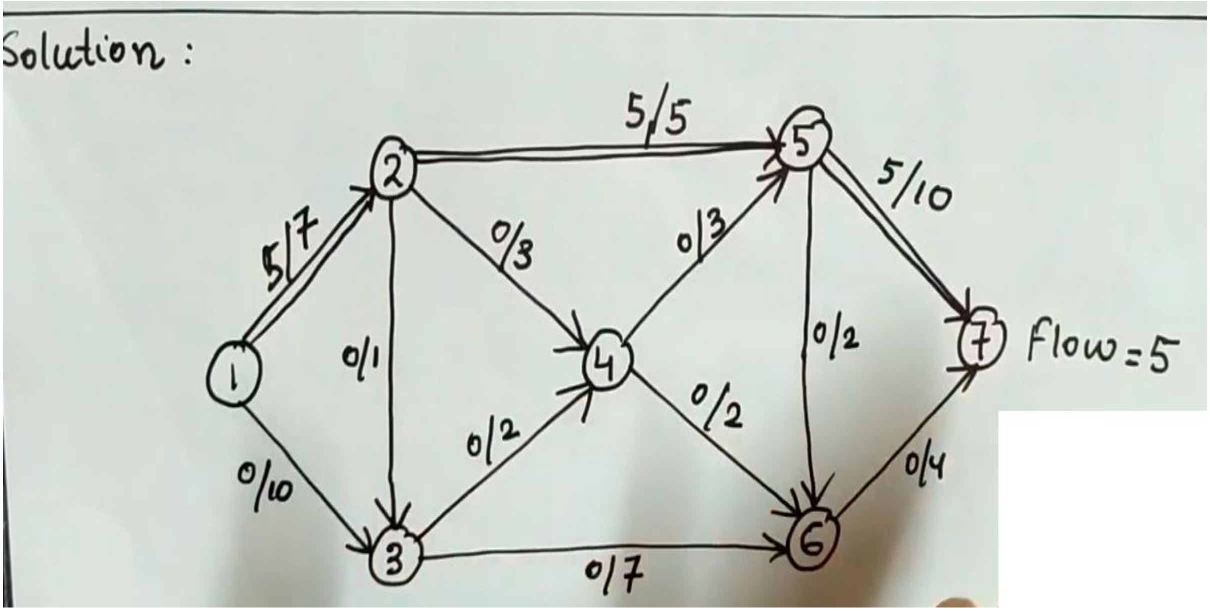








Augumenting path	Bottleneck capacity
1-2-5-7	5
1-3-6-7	4
1-2-4-5-7	2



Continuesxxxxxxxxxxxxxxxxxxxxx