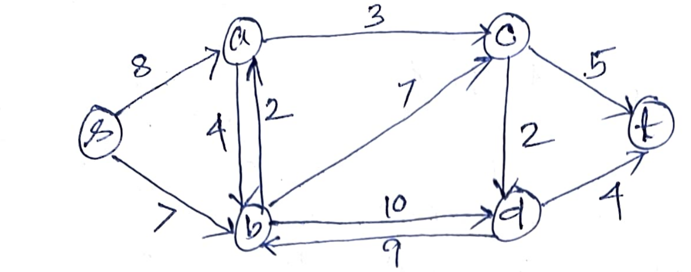
1 (a)



(b) possible cut-sets;

1. $(\{s\}, V-\{s\})$: $C = 15$
2. $(\{s,a\}, \{b,c,d,t\})$: $C = 7+4+3 = 14$
3. $(\{s,b\}, \{a,c,d,t\})$: $C = 8+2+10+7 = 27$
4. $(\{s,c\}, \{b,a,d,t\})$: $C = 8+7+5+2 = 22$
5. $(\{s,d\}, \{a,b,c,t\})$: $C = 8+7+4+9 = 28$
6. $(\{s,a,b\}, \{c,d,t\})$: $C = 3+7+10 = 20$

⋮

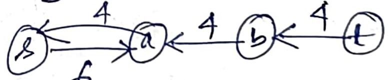
16. $(\{s,a,b,c,d\}, \{t\})$: $C = 5+4 = 9$. (minimum)

Apart from these s-t-cuts many other cuts are possible in general

(c) FALSE.

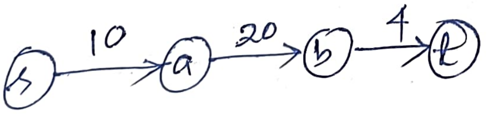
The edges across the cut don't solely decide the possible augmentations. If we increase the capacity of the cross edges of a minimum cut, there might be any other edge which emerges ~~as the~~ with the bottleneck capacity and decides the possibility of augmentation and hence can decide the max-flow. This can be visualised from the following counter example, —

after FF; —

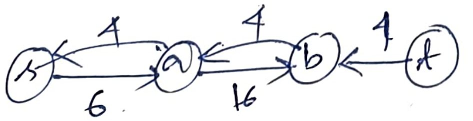


$\boxed{f=4}$ cut $(\{s,a\}, \{b,t\})$

If we increase the capacity of the edge (a,b) which crosses the min^m cut, —



, the flow still remains 4. (no change)



1 2.(a) Given n groups to be assigned to 'm' sessions such that the lo[ad] on any group shouldnot exceed 'l_i' for group i. The assignment depends on the availability of groups for sessions.
We can map this problem as a maximum flow problem.
The modelling of the problem as max^m-flow problem as follows:—

Create a flow network $G=(V,E)$ where,—

$$V = \{s_1, s_2, \ldots, s_m\} \cup \{g_1, g_2, \ldots, g_n\} \cup \{s, t\}$$

s ↓ source, t ↳ sink

$E = E_1 \cup E_2 \cup E_3$

$E_1 = \{(s, s_i) : \forall i,\ 1 \le i \le m\}$

$E_2 = \{(s_i, g_j) : \forall i, j$ where $1 \le i \le m$ and $1 \le j \le n$ and group j is available for session $s_i\}$

$E_3 = \{(g_j, t) : \forall j$ where $1 \le j \le n\}$

Assign capacities of the edges as follows:

$$C(e) = \begin{cases} 1 & \text{if } e \in E_1 \\ 1 & \text{if } e \in E_2 \\ l_j & \text{if } e \in E_3 \end{cases}$$

The flow can be viewed as the load assigned to the groups in terms of no. of sessions or hours (as each session last for 1 hours).
Now, we can define the instance of the maximum flow problem as $\langle G, s, t, C \rangle$

objective:— $\quad$ maximize $\displaystyle\sum_{e \text{ out of } s} f(e)$

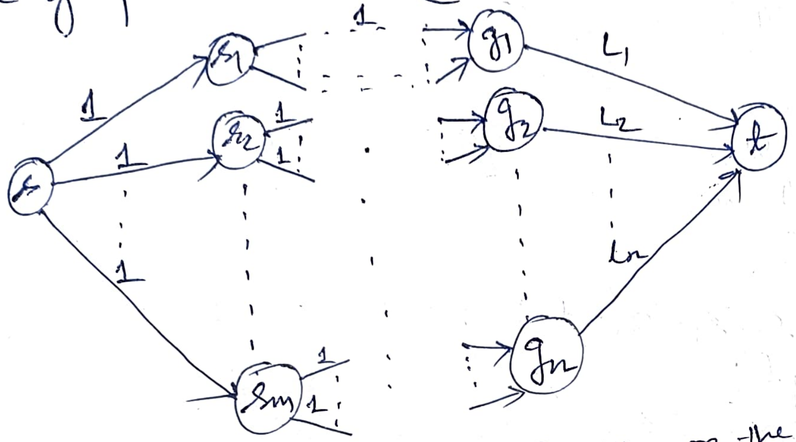
subject to: $\quad 0 \le f(e) \le C(e) \quad \forall e \in E$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e) \quad \forall v \in V - \{s,t\}$$

(b) As we have modelled the given problem as an instance of the Maximum Flow problem in the previous question (Q.2.a), we can solve it using the FF algorithm.

Given instance of max-flow problem as $\langle G, s, t, C \rangle$

The graph will look like the following –



when we apply the FF algorithm on the above graph, –

each augmenting path will ~~to~~ have a general form as –

$(s) - s_i - g_j - (t)$.

and every augmenting path will have a bottleneck capacity = 1

→ The number of augmentation will depend on the availability of groups for sessions (i.e. no. of outdegrees of s_i nodes):

→ On each augmentation, one (s, s_i) edge and one (s_i, g_j) edge will saturate with the flows and each will incur a reverse edge (s_i, s) and (g_j, s_i) with unit capacity.

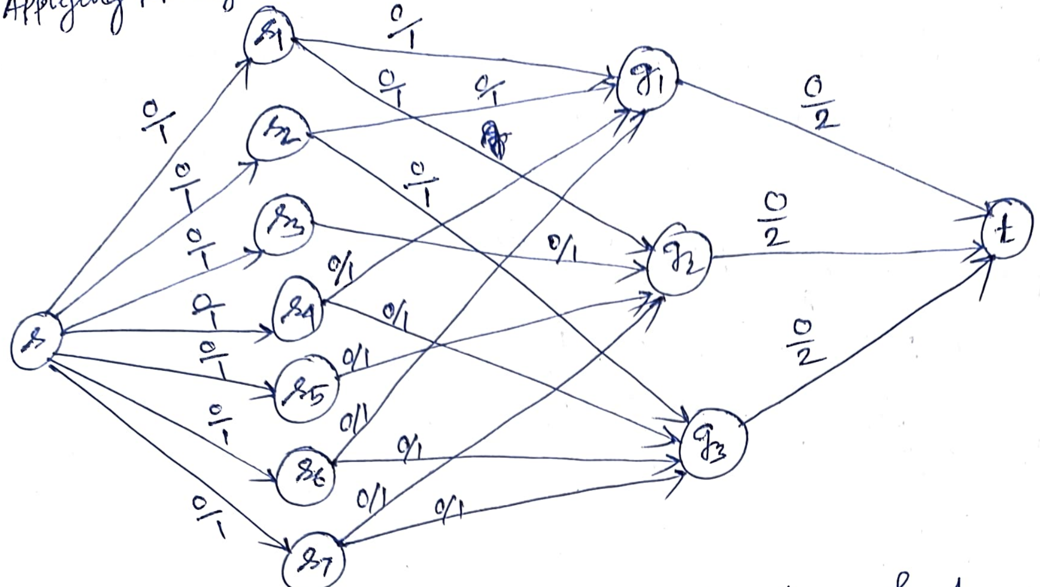
→ The max-flow value = No. of augmentations possible.

→ The FF algo will terminate if no augmentation is possible.

→ After the termination of FF algo, we can find the assignments from the final flow network. The assignment M can be defined as;

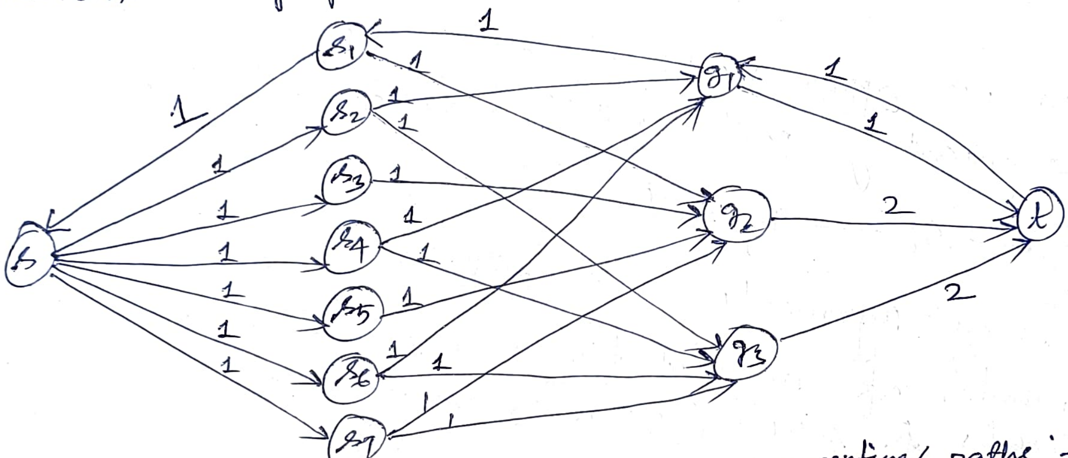
$M = \{ (s_i, g_j) : 1 \leq i \leq m$ and $1 \leq j \leq n$ such that, there is a flow of 1 unit between s_i and $g_i \}$

(c) Applying FF algo:-



Augmenting path: $s - s_1 - g_1 - t$: $b = 1$ $f = 1$.

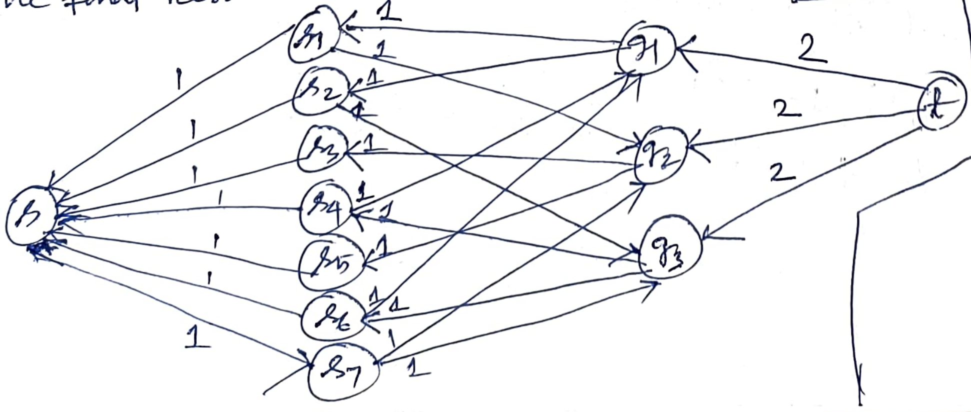
The residual graph:



Similarly, we can find the following augmenting paths:-

$s - s_2 - g_1 - t$	$b = 1$	$f = 1+1 = 2$
$s - s_3 - g_2 - t$	$b = 1$	$f = 2+1 = 3$
$s - s_4 - g_3 - t$	$b = 1$	$f = 3+1 = 4$
$s - s_5 - g_2 - t$	$b = 1$	$f = 4+1 = 5$
$s - s_6 - g_3 - t$	$b = 1$	$f = 5+1 = 6$

The final residual n/w will be as follows:-

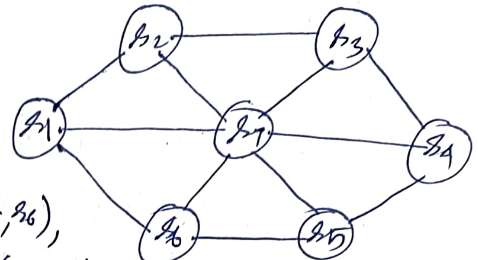


- one session remains unassigned (s_7)
- The assignment consists of order pairs between session s_i and group g_j as (s_i, g_j).

The result will contain the assignment as follows: $\{(s_1, g_1), (s_2, g_1), (s_3, g_2), (s_4, g_3), (s_5, g_2), (s_6, g_3)\}$

③ⓐ To find the decision set-cover problem instance $\langle U, \{s_1, \dots, s_m\}, k \rangle$ we need to define each entity of the instance.

Given VC instance $\langle G, K \rangle$



SC instance $\langle U, \{S_1, \dots S_m\}, k \rangle$

$U = E.G$ = edges of G.

$= \{(s_1,s_2),(s_2,s_3),(s_3,s_4),(s_4,s_5),(s_5,s_6),$
$(s_6,s_1),(s_1,s_7),(s_2,s_7),(s_3,s_7),(s_4,s_7),(s_5,s_7),(s_6,s_7)\}$

$S_1 = \{(s_1,s_2),(s_1,s_6),(s_1,s_7)\}$ $S_5 = \{(s_4,s_5),(s_5,s_6),(s_5,s_7)\}$

$S_2 = \{(s_1,s_2),(s_2,s_3),(s_2,s_7)\}$ $S_6 = \{(s_5,s_6),(s_6,s_7),(s_1,s_6)\}$

$S_3 = \{(s_2,s_3),(s_3,s_4),(s_3,s_7)\}$ $S_7 = \{(s_1,s_7),(s_2,s_7),(s_3,s_7),(s_4,s_7),(s_5,s_7),$
$(s_6,s_7)\}$.

$S_4 = \{(s_3,s_4),(s_4,s_5),(s_4,s_7)\}$

K is the same i.e. $K = 4$.

ⓑ Vertex-Cover $\leq_p$ Set Cover.

Vertex-cover(G, K)

construct a universal set $U = \{(u,v) : \forall (u,v) \in E_G\}$

construct a collection S of subsets as $S = \bigcup_{v \in V.G} \{S_v\}$

where S_v = set of all edges incident on vertex v.

return Set-cover(U, S, K)

ⓒ In the given problem, the Minimum VC = $\{s_1, s_3, s_5, s_7\}$

Minimum SC = $\{S_1, S_5, S_3, S_7\}$

Maximum SC = $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$

4ⓐ FALSE.

Given $X \leq_p Y$ where $Y \in NPH$.

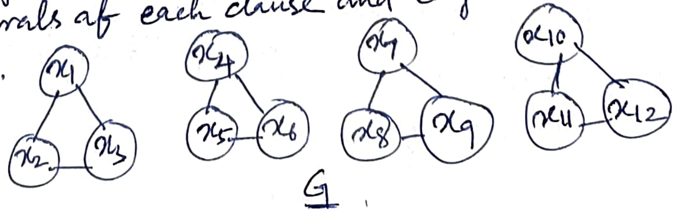
A problem is NPH if every problem in NP can be polynomially reducible to it. This is a property of Y here. It doesnot say anything about X. So, X can be in any ~~problem~~ ~~els~~ complexity class such as P, NP, NPC, or exclusive NPH as well.

ⓑ Given a 3-SAT formula $\phi(x) = (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \wedge$
$(x_{10} \vee x_{11} \vee x_{12})$

To reduce it into an instance of IS problem, ($\langle G, K \rangle$) we need to define G and K.

- We can construct a graph with $3 \cdot K$ (K = no. of clauses) nodes and an edge between the literals of each clause and edges between conflicting literals (i.e. x_i and $\bar{x}_i$).

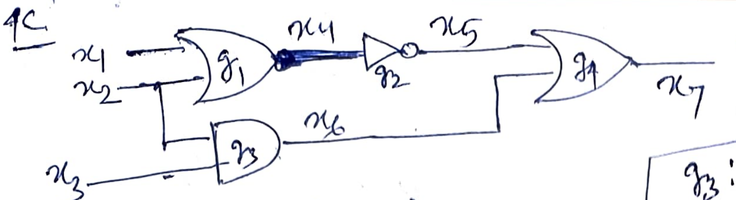
→ In this graph, the IS =
$\{x_1, x_4, x_7, x_{10}\}$



G

SC.instance
$\langle G, 4 \rangle$

1c



g_2: $\phi_2 = x_5 \Leftrightarrow \overline{x_4}$
$= (\overline{x_5} \vee \overline{x_4}) \wedge (x_5 \vee x_4)$

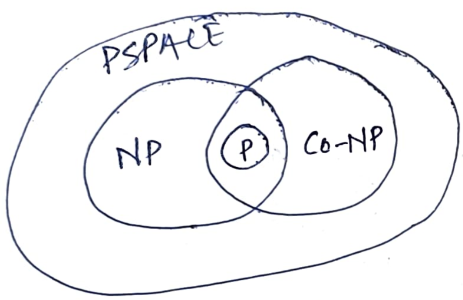
g_3: $\phi_3 = x_6 \Leftrightarrow (x_2 \wedge x_3)$
$= [\overline{x_6} \vee (x_2 \wedge x_3)] \wedge [x_6 \vee (\overline{x_2 \wedge x_3})]$
$= (\overline{x_6} \vee x_2) \wedge (\overline{x_6} \vee x_3) \wedge (x_6 \vee \overline{x_2} \vee \overline{x_3})$

g_1: $\phi_1 = x_4 \Leftrightarrow (x_1 \vee x_2)$
$= (\overline{x_4} \vee x_1 \vee x_2) \wedge ((\overline{x_1 \vee x_2}) \vee x_4)$
$= (x_1 \vee x_2 \vee \overline{x_4}) \wedge ((\overline{x_1} \wedge \overline{x_2}) \vee x_4)$
$= (x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_4) \wedge (\overline{x_2} \vee x_4)$

g_4: $\phi_4 = x_7 \Leftrightarrow (x_5 \vee x_6)$
$= (x_5 \vee x_6 \vee \overline{x_7}) \wedge (\overline{x_5} \vee x_7) \wedge (\overline{x_6} \vee x_7)$

$\phi = x_7 \wedge \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$.
$= (x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_4) \wedge (\overline{x_2} \vee x_4) \wedge (\overline{x_5} \vee \overline{x_4}) \wedge (x_5 \vee x_4) \wedge (x_2 \vee \overline{x_6}) \wedge$
$(x_3 \vee \overline{x_6}) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_6) \wedge (x_5 \vee x_6 \vee \overline{x_7}) \wedge (\overline{x_5} \vee x_7) \wedge (\overline{x_6} \vee x_7)$

5.a



(b) Given a graph $G(V,E)$ with positive profit/weight values associated with each vertex and a +ve integer K. Two players A and B select the vertices alternately to acquire the profits associated with them, with a condition that, if a vertex is selected by any player, its adjacent vertices can not be selected by either of the players. Can the first player A makes the 1st selection, can the second player B achieves a profit of at least K?

(c) We can transform the given Q-SAT problem into 3SAT problem by replacing all the '$\forall$' quantifiers with '$\exists$' quantifiers. then the problem becomes :-

whether there exist a choice for x_1 so that ~~for~~ there is a choice for x_2 so that there is a choice for x_3 and so on so that ϕ is satisfiable?

i.e. $\exists x_1 \exists x_2 \exists x_3 \ldots \exists x_{n-2} \exists x_{n-1} \exists x_n \; \phi(x_1, x_2, \ldots, x_n)$ is satisfiable?