ঢাকা বিশ্ববিদ্যালয়

দ্বিতীয় বর্ষ প্রথম সেমিস্টার পরীক্ষা, ২০২০

বিষয় সফটওয়্যার ইঞ্জিনিয়ারিং পত্র/ কোর্স নং CSE-301

কেন্দ্র IIT (অনলাইন) রোল নম্বর 1122

ঢাকা বিশ্ববিদ্যালয় রেজিঃ নম্বর 2018225587 শিক্ষাবর্ষ 2018-19

তারিখ 09.08.2021

Q 3

LCS of <1, 0, 0, 1, 0, 1, 0, 1> and <0, 1, 0, 1, 1, 0, 1, 1, 0>

Lets define lcs [i][j] :- as the longest common Subsequence till index i of first string and j of second string.

$$dp[i][j] = dp[i-1][j-1]+1 \quad [\text{if } s1[i] = s2[j]]$$

else, $dp[i][j] = \max(dp[i-1][j],\ dp[i][j-1]);$

for the given strings :-

S1 ↓	S2 →	.	0	1	0	1	1	0	1	1	0
	i \ j	0	1	2	3	4	5	6	7	8	9
	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	1	1	0	1	1	0
0	2	0	1	1	2	2	2	2	2	2	2
0	3	0	1	1	2	2	2	3	3	3	3
1	4	0	1	2	2	3	3	3	4	4	4
0	5	0	1	2	3	3	3	4	4	4	5
1	6	0	1	2	3	4	4	4	5	5	5
0	7	0	1	2	3	4	4	5	5	5	6
1	8	0	1	2	3	4	5	5	6	6	6

LCS is length 7.

and string is: ~~010101~~. 100110

2

Amount of money C with minimum number of coin :- Given $v_1 < v_2 < v_3 \dots < v_n$ [coin values]

```
# include <bits/stdc++.h>
using namespace std;
int   minimumCoin (int [], int, int)
int main(){

    int coins[] = {V1, V2, .... Vn};
    int n = size of(coins)/sizeof(int);
    int V= C; // C is target Value.
    int res= minimumCoin(coins, n, V);
}    cout<< res<< endl;
```

(P.T.O)

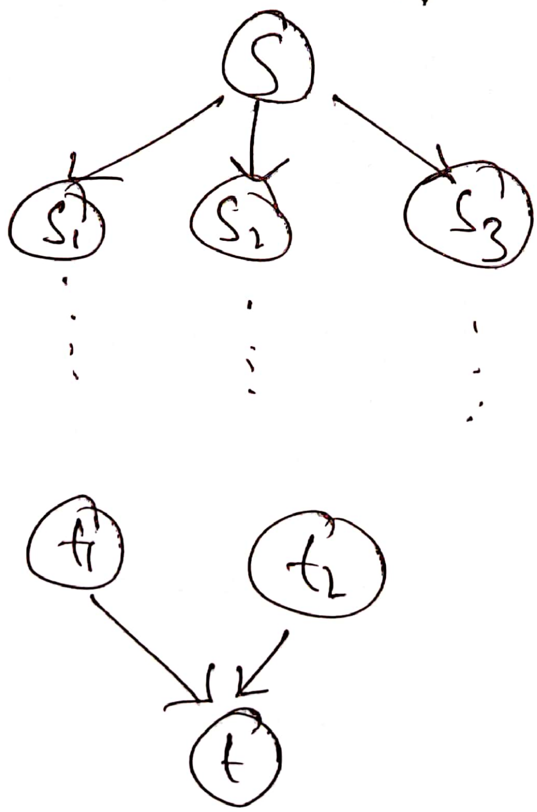
```
int minimumCoin( int coins[], int m, int target)
{
    int dp [target+1];
    dp [0] = 0;
    for (int i=1; i< =target ; i++)
        dp[i]= INT_MAX;

    for (int i= 1; i<= target; i++){
        for (int j =0; j<m; j++){
            if (coins[j] <= i){
                dp [i]= min(dp [i- cons[j]] +1,
                                dp[i]);
            }
        }
    }
    return dp[target];
```

4(a).

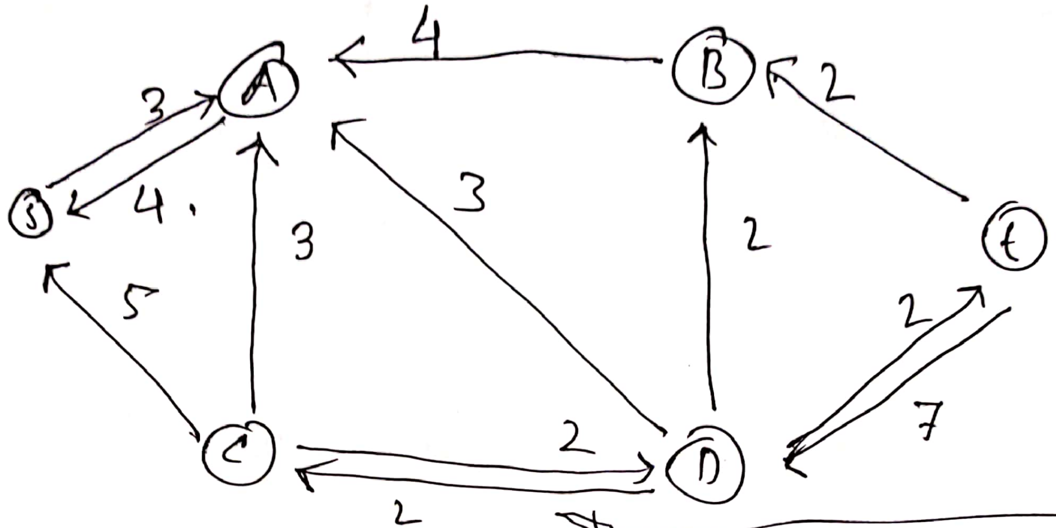
Sources (S_1, S_2, S_3) and Sinks ($t1, t2$)

It can be converted into a maximum flow problem if we make a SUPER SINK. with the capacity of the total capacity of the sinks and a "super source" following the same for all sources.



4(b)

residual Network of the graph:-



Hence, $\boxed{\text{the maximum flow is 9}}$

Its already in maximum flow. Since, there is no path augmenting from source to sink.

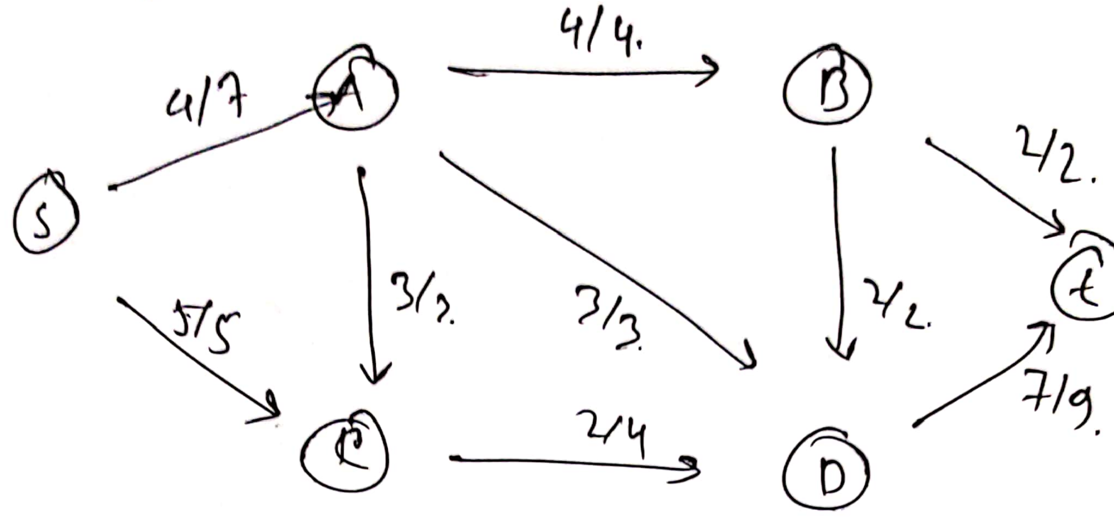
We know, from the rule of flow conservation rule, that for each vertex, the ~~input~~ ~~flow~~ summation of incoming flow and the summation of outgoing flow should be equal.

outgoing → $\sum f(u,v)$ $v \in V \setminus u$.

$\sum f(u,v) = \sum f(v,u)$.

incoming → $\sum f(v,u)$ $v \in V \setminus u$

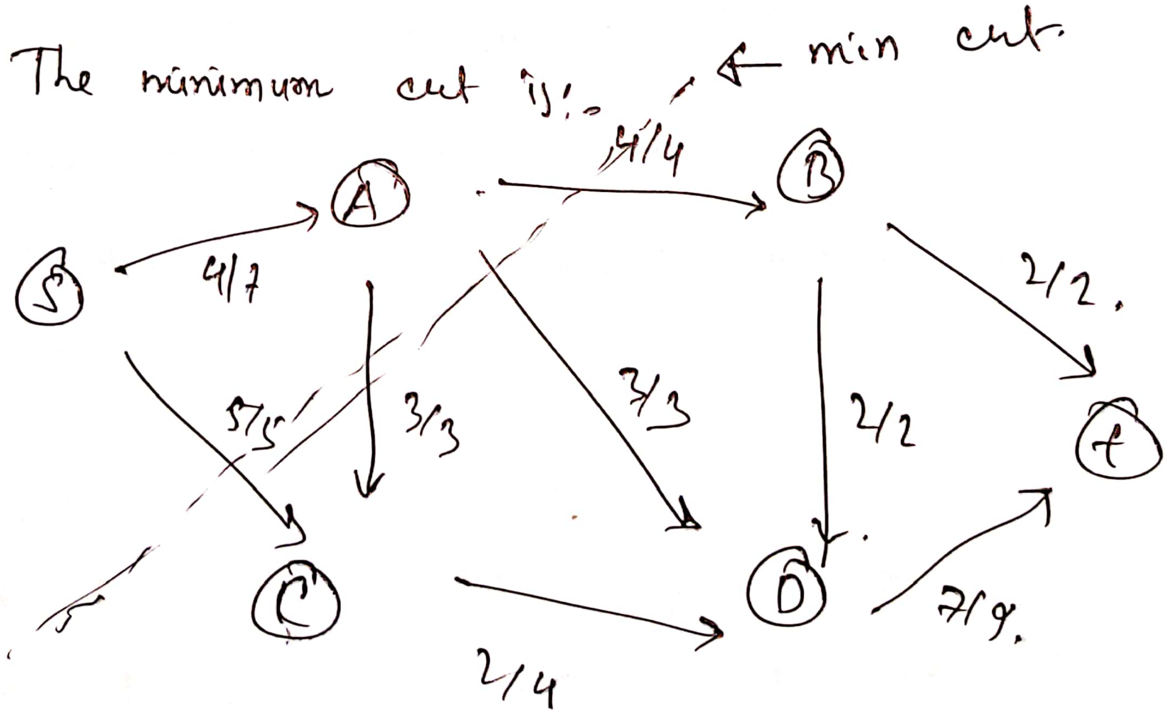
But from the given graph :-



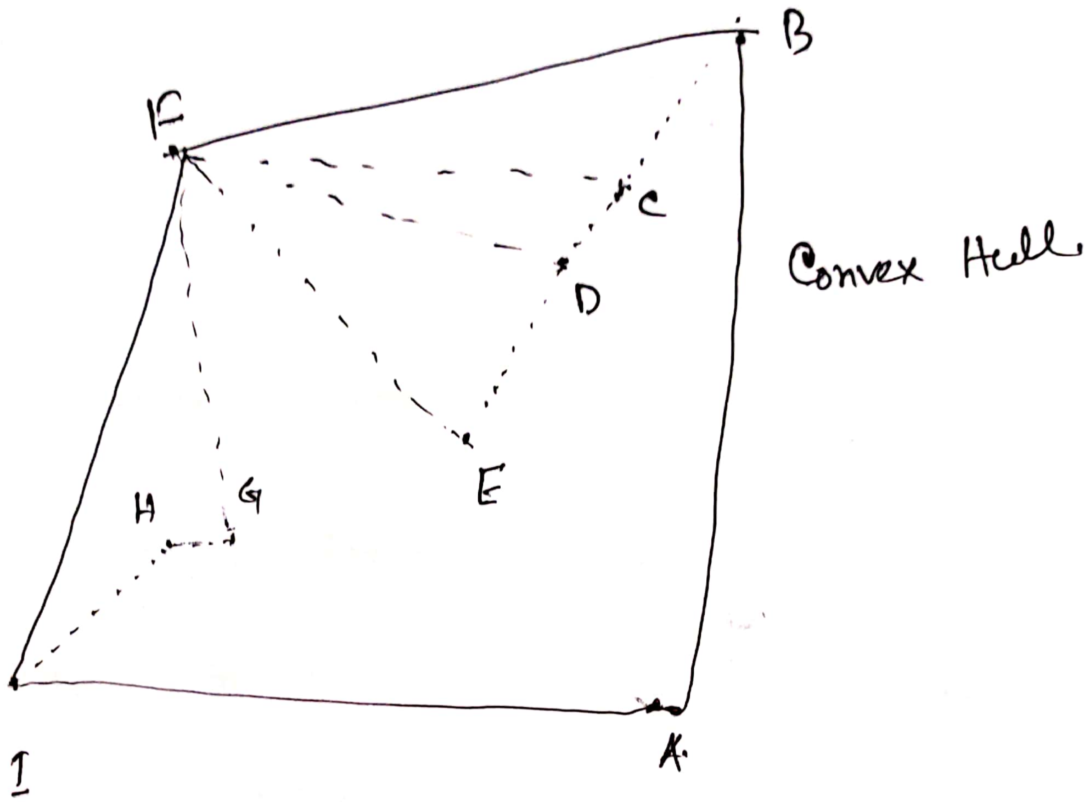
in C, the incoming flow is = 8
outgoing flow is = 2

which contradicts with the property of flow conservation.

The minimum cut is :-



5 a



Here, we pop from stack for non-right move,

first we push, A, I, H to stack.

* then, G, right move, push.
* F makes left move, pop G, again make left, pop, H, push F.

* . E makes right move, push.
* D " left move, pop, E . push D.
* C " " " , " D, push C
* B " " " " C, push, B.

*. B makes left move, push. B.

So, convex hull, (A, L, F, B)

5(b)

i) Polynomial time Problems:-

The problems which have an algorithm to be solved in polynomial time are called Polynomial problems.

Here complexity should be, $O(n^k)$

k is a constant.

ii). NP Hard:-

A problem B is NP hard if.

for all NP-Complete Problems A, A can be reduced to B.

$A \leq_p B$ [$B \in NPC$].

(iii) NP- Completeness:-

A problem B is NP-Complete if:-

(i) It has no polynomial time ~~of~~ ~~to~~ ~~has~~ solution.

thus, $B \in NP$. AND,

(ii) for all problems $A \in NP$, A can be reduced to B.

$A \leqslant_p B$.

If a problem follows both property, its NP complete.

1(a)

Given, LPP,

Maximize, $z = 5x_1 - 2x_2$.

$-x_1 + x_2 \leq -2$.

$2x_1 + 3x_2 \leq 5$

$x_1, x_2 \geq 0$

Introducing surplus variables, $S_1, S_2 \geq 0$.

Maximize $z = 5x_1 - 2x_2 + 0.S_1 + 0.S_2$.

~~Standard form.~~ $-x + x_2 - S_1 = -2$.

$2x_1 + 3x_2 - S_2 = 5$

$x_1, x_2, S_1, S_2 \geq 0$.

Associated dual is,

~~Maximize,~~ Minimize $z = -2\omega_1 + 5\omega_2$.

$-\omega_1 + 2\omega_2 \leq 5$

$\omega_1 + 3\omega_2 \leq -2$.

$\omega_1, \omega_2 \geq 0$

1(b)

let, let, ~~sent fro~~

Number of employee,

sent from little rock to ~~urbana~~ louis. $= x_1$

" " " " to detroit $= x_2$

" " Urbana to louis $= x_3$.

" " " to detroit $= x_4$

Cost is : $z = 400x_1 + 200x_2 + 100x_3 + 200x_4 + 280$

[minimize].

Constraints : $x_1 + x_2 \leq 6$.

$x_3 + x_4 \leq 6$.

$x_1 + x_3 \geq 5$

$x_2 + x_4 \geq 4$.

$x_1, x_2, x_3, x_4 \geq 0$