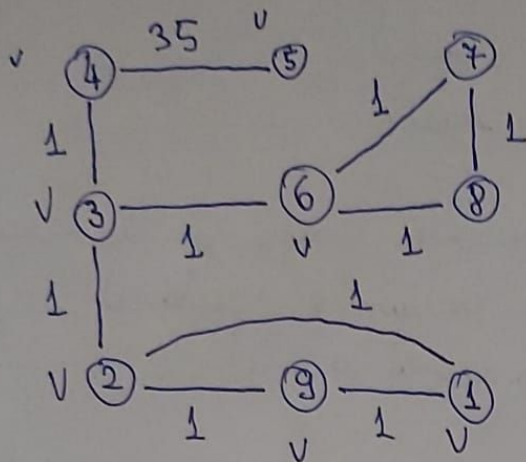


SUBIECTUL I

Genia
Gabriel



1) m. critice : 2, 3, 4, 6

2) m. critice : 3-4

3-6

3-2

4-5

3) dfs(3):

- notăm V_3 ca vizitat

- ~~treccm în V~~

- Verificăm V_2 dacă a fost vizitat (nu a fost)

- treccm în V_2

- notăm V_5 vizitat (+verif V_2 vizitat (da))

- verif $V_8 \dots$

- treccm în V_9

(tote vecinii lui V_9 sunt vizitati \Rightarrow ne întorcem în V_1)

- notăm V_2 vizitat
- Verificăm V_1 vizitat (nu)
- treccm în V_1

(tote vecinii lui V_1 sunt vizitati \Rightarrow ne întorcem în V_2)

(V_2 ,
 V_3)

- Verif $V_4 \dots$ (nu)

- treccm în V_4

- notăm V_4 vizitat

- Verificăm V_5 vizitat (nu)

- treccm în V_5

- notăm V_5 vizitat

- ne întorcem în V_4

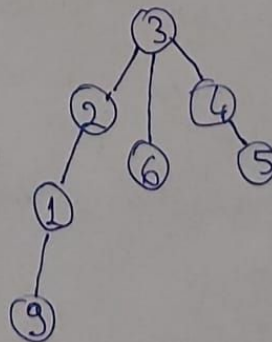
- în V_3

- Verif. V_6 vizitat (nu)

- treccm în V_6

- notăm V_6 vizitat

Arbore DFS(3)



4) - am ales muchia [4-5] dintre muchiile critice

- am pus ponderea 1 pe restul muchiilor

- 1 - am pus pe [4-5] diferența 42-7

5) a = examen
b = setanta

b = restante

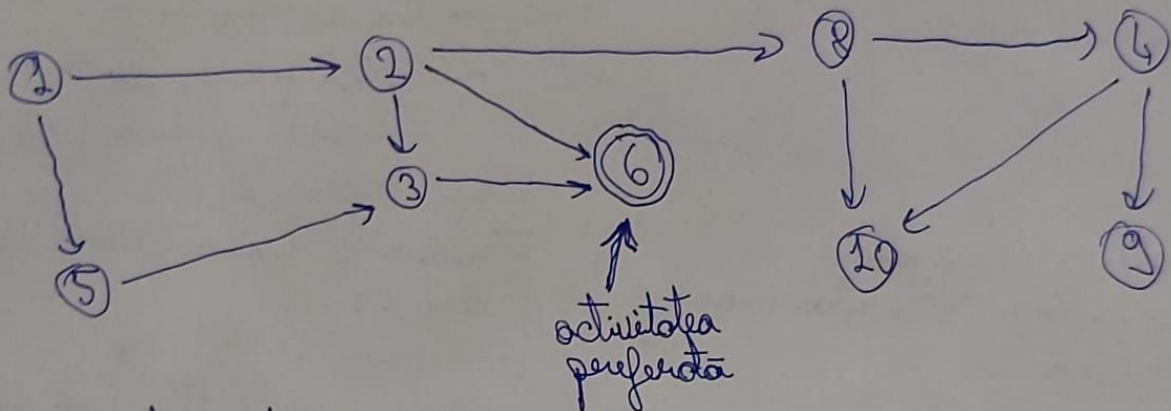
1. steigern positive 1, 3, 4 \Rightarrow "can"
2. adäquatem 9 [e], s, t, [am], t, a

2. adāṅgām a [e], s, t, [am], t, a

\Rightarrow distanța solitare = $3 + 5 = 8$ (operații)

↓
stergari

6) Această problemă se poate rezolva folosind algoritmul de sortare topologică, ordinea acțiunilor putând fi modelată ca graf orientat neponderat aciclic (DAG)



o sentare topologica bidă :

1, 5, 2, 3, 6, ...

complexitatea sortării topologice pe DAG-uri este $O(n + m)$, unde $\begin{cases} n - \text{nr. de noduri} \\ m - \text{nr. de muchii} \end{cases}$, fiind cel mai eficient algoritm pentru scenariul dat.