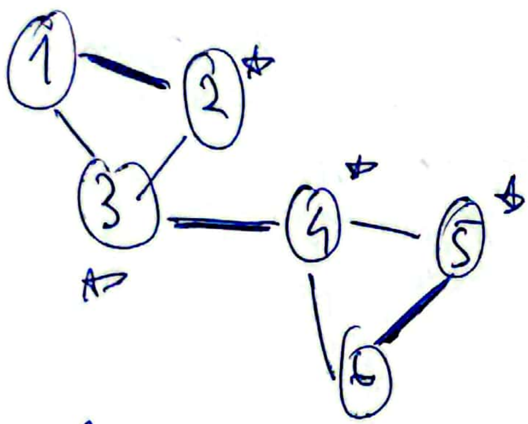
# PAG - Curs idk..4?

## Algoritmi de 2-aproximare pt vertex cover.

**Def Vertex Cover:** Se dă un graf neorientat. Să se găsească o submulțime de noduri $S \subseteq V(G)$ de cardinalitate minimă a.î. $(\forall)\ (a, b) \in \underbrace{E(G)}_{\text{mulțimea muchiilor}}$ să avem $a \in S$ sau $b \in S$ sau ambele.

**Ex:**



$S = \{2, 3, 4, 5\}$

**Def Cuplaj:** Într-un graf neorientat $G = (V, E)$ este o mulțime de muchii $M \subseteq E$ a.î. $(\forall)\ e, e' \in M$ nu au noduri în comun.

**Def Cuplaj maximal:** Un cuplaj $M$ într-un graf $G$ este maximal dacă nu există $e \in E - M$ a.î. $M \cup \{e\}$ să fie un cuplaj.

**Def Cuplaj maxim:** Un cuplaj $M$ într-un graf neorientat $G$ este maxim dacă $(\forall)\ M' \subseteq E$ un cuplaj, avem $|M| \geq |M'|$

Lemă: $(\forall)$ G graf, $G=(V,E)$, Vertex Cover este cel puțin cât un cuplaj maximal

Demonstrație:

P.P. $\exists$ un cuplaj maximal M a.î. $\underbrace{|V_c|}_{\text{Mărimea Vertex cover}} < |M|$

atunci $\exists$ $(a,b) \in M$ a.î. $(a,b) \notin V_c$

Contradicție! cu def. problemei; Ar însemna că $(a,b)$ nu e acoperită.

Algoritm de 2 aproximare pt vertex cover.

1. Alegem un cuplaj maximal M
2. Adăugăm în $V_c$ ambele capete ale muchiilor din M

Teoremă: Algoritmul de mai sus e o 2-aproximare

$$ALG = 2 \cdot |M| \leq 2 \cdot OPTIM$$

Dem: P.p RA $\rightarrow \exists$ $(a,b) \in E$ a.î. $(a,b) \notin V_c$

$\Rightarrow M \cup (a,b) =$ cuplaj contradicție pt că M este maximal.

Când dai un alg de aproximare te întrebi două lucruri:

1. Este analiza calculului precisă („tight") -> Adică alg. poate se descurcă mai bine? (Ca să răspunzi ~~în~~ cauți un exemplu care dă fix în cazul nostru 2 × OPT

ALG = {1, 2}
OPT = {1} => e tight.)

2. Putem obține un algoritm mai bun? (de 2 aproximare?)
În cazul nostru

În cazul nostru -> nu se știe :)

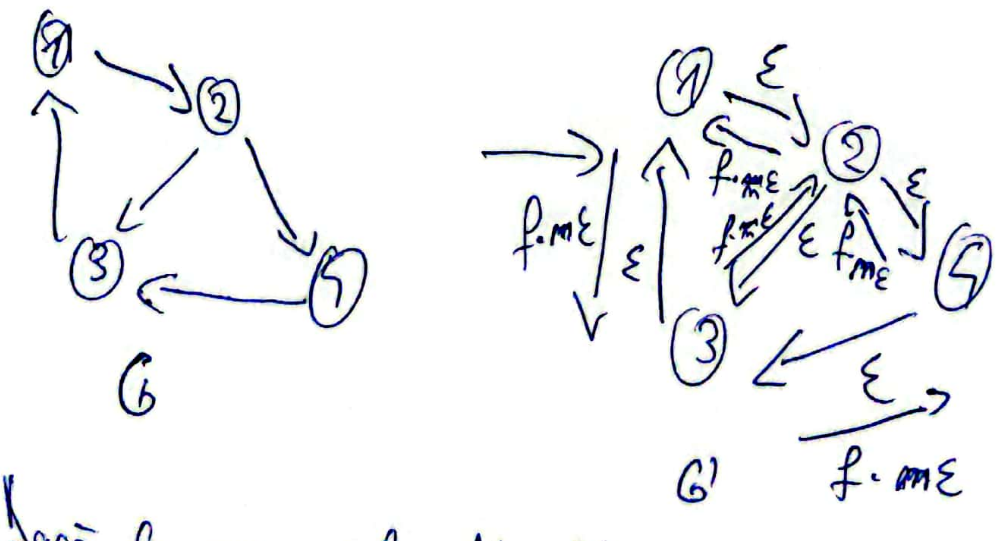
# Problema TSP

Se dă un graf orientat complet $G=(V,E)$ și o funcție $w: E \to \mathbb{R}_+$. Să se găsească un tur care vizitează fiecare nod o singură dată și are cost total minim.

<u>Teoremă</u>: Pt/ orice funcție $f$ calculabilă în timp polinomial, dacă problema TSP admite un algoritm de $f$-aproximare atunci $P=NP$.

<u>Demonstrație</u>: Reducție de la problema ciclului Hamiltonian:

☒ Dându-se un graf $G=(V,E)$, construim o instanță a TSP $(G', w)$ a.î. putem decide în timp polinomial dacă $G$ este hamiltonian dacă avem o $f$. aproximare pentru TSP.

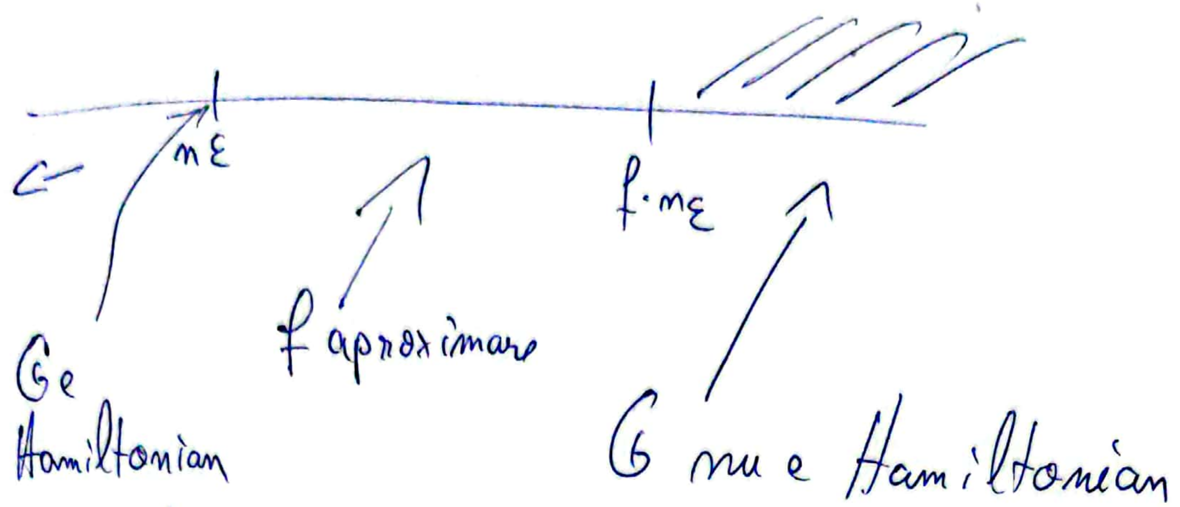



Dacă $G$ are ciclu Hamiltonian $\Rightarrow$ cost $= \cancel{m\varepsilon}\ m\varepsilon$

Dacă $G$ nu are ciclu H—n— $\Rightarrow$ cost $\geq f \cdot \cancel{m\varepsilon}\ m\varepsilon$

~~Nu~~

=>




=> Putem din schema asta să decidem dacă G e sau nu Hamiltonian din aproximarea TSP
=> no se puede pt că problema ciclului Hamiltonian e NP Hard.

## Problema Max-3SAT:

Se dă o formulă $\phi$ în care fiecare clauză are exact trei literali, să se găsească un assingment care satisface un număr maxim de clauze.
7/8 aproximare.
Upper Bound = m -> nr de clauze.
Cum putem satisface 7/8 din clauze?
Setăm $x_i$ = true aleator cu probabilitate 1/2.
P(C să fie satisfăcută) = 1 - P(C să nu fie satisfăcută)

$= 1 - P(x_1 = \text{false} \wedge x_2 = \text{false} \wedge x_3 \wedge \text{false})$

$= 1 - \ldots = 1 - \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = 1 - \frac{1}{8} = \frac{7}{8}$ GG.