Complexité.

Examen l'année dernier 2021/2022.

1)

Q1) Cours / 2-SAT ≤ SAT.

1 ⇒ P = NP / En pas déduire Tout les 4 propositions.

Q2) A un langage appartient à P.

1 – Vrai.
2 – Vrai.
3 – Non, Manque d'information.
4 – Non, on peut pas dire P = NP avec une machine Non-determinis

NP_complete = NP-diff + NP.
$\forall B \in NP \; B \leq A$.

Q3.)

1 – Vrai, si A NP-complet alors C est NP-complete.
2 – Vrai.
3 – Non. à cause de la direction.
4 – N.

Si C est facile alors B, A il est aussi.

2)

1) NP ≠ coNP, alors P ≠ NP.

$P \subseteq NP$
$P \subseteq coNP$

$P = coP$.
$P = NP$ du coup c'est
$coP = coNP$ égalité.

3) Appartenance à la classe NP.

~~[illegible]~~ il faut avoir un algo-non-determinist
ou un vérificateur déterminist.

Dem 3 - SAT :

$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee x_5) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_5)$

	1	2	3
1	1	-2	3
2	-1	-2	4
3	2	-3	5
4	-2	4	5

π

$x_1 = 1$.
$x_2 = 0$
$x_3 = 0$
$x_4 = 0$
$x_5 = 0$.

Il doit pas y avoir 3 ~~clause~~ letteraux vrai dans une même formule.

Formule conjonctif.

2)

Algo DOV 3-SAT(F) :
    X = tableau de long F.n   O(1)
    pour i = 1 à F.n faire O(n)
        X[i] = devine (0, 1).
    pour i = 1 à F.m faire . O(m)
        nsat = 0.
        si F.c[i][1] > 0 et X[F.c[i][1]] = 1 ou
        F.c[i][j] < 0 et X[-F.c[i][j]] = 0)
            nsat = nsat + 1 ; incrémentation de nsat.
        if nsat < 1 ou nsat > 2 alors
            rejeter   => 1 du coup elle est pas satisfait ;
                      => si est superieur à 2 alors faut
    accepter.   O(m)   parceque elle accepte pas la
                       condition.

algo non det.

un algorithme qui vérifie si une formule est satisfaite

(2) Le complexité :
temps total = $O(m+n)$. TOu.
pire des cas = $\Theta(m+n)$. theta

cette algo ~~nd~~ est non-déte

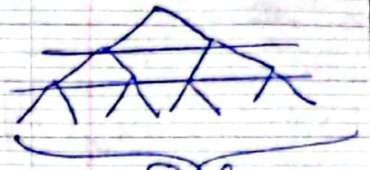
Scanné avec CamScanner

code des entier de 1 à n $O(n(\log(n)))$

$3 m \log n$.

$O(m) \subseteq O(m \log n)$.

$3m$



2^m = choisir dans.

2.)

L'algo verificateur.

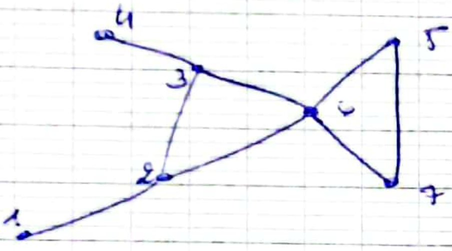
Le même code, en ajout X dans l'entée.
et supprimer la partie X.

Partition en clique :

$m \in O(n^3)$.

$\binom{n}{3} 2^3 \in O(n^3)$

1.

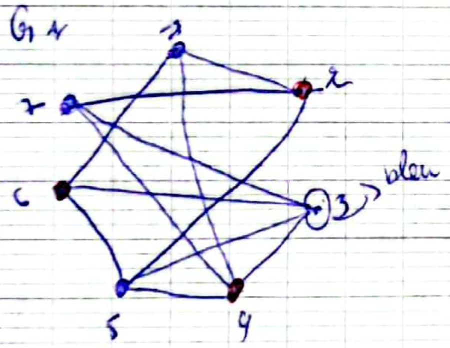


$\{2, 3, 6\}$ une petit clique de G.

3-coloration à Partition en clique, avec une réduction manyone :

Q1) Trouver une 3-coloration des sommet de G_1, et une Partition G_2 :

G_1



$G2$ = graphe complémentaire de G_1.

Le graphe forme 2 clique :

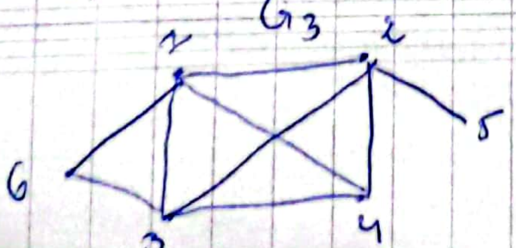
$\{1, 5, 7\}$, $\{2, 4, 5\}$ $\{3\}$

Q2) Une partition de G_2 en troi clique :

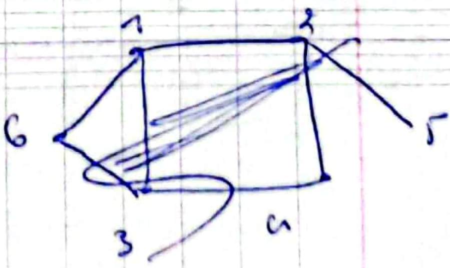
$\{2,4,6\}$, $\{1,3\}$, $\{5,7\}$ une autre partition pour le graphe G_2.

Q3) 6 sommet qui admet pas 3-colloration.

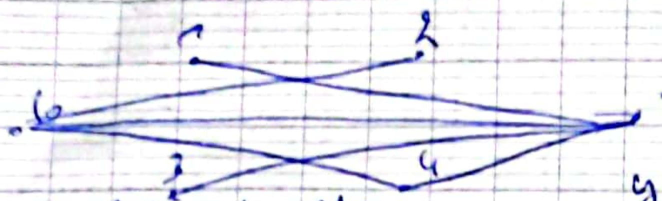
G_3



$\Rightarrow$



complémentaire.



$k = 3$

$\{4,5,6\} / \{1\} / \{2\} / \{3\}$

y a pas pb d'arrête, le graphe n'est 3_coloration.

(Q4) un algorithme non-det en temps polynomiale.

```
algo Reduction(G1):
    G2 = graphe de G1.n sommets
    pour i = 1 a G1.n faire
        pour j = 1 a G2.n faire
            Si i ≠ j et G1.M[i][j] = 0 alors
                G2.M[i][j] = 1
            sinon
                G2.M[i][j] = 0
    K = 3
    renvoy (G2, K)
```

temps $= O(n^2)$

Q5) $G_1 \in 3COL \Rightarrow (G_2, K) \in$ PART_cliques.
$(S', A') \quad 3$

(S_1, S_2, S_3)
$S_1 \cup S_2 \cup S_3 = S$
$S_i \cap S_j = \emptyset$ // Sinon on aura 2 couleurs pour un seul sommet qui est impossible.

$\forall u, v \in S_j$, $u \neq v$ $\{u, v\} \in$

Q6) Le cas contraire :

$S_1, S_2, S_3 \subseteq S$ $\qquad S_1 \cup S_2 \cup S_3 = S$
$\forall u, v \in S_i$ $\qquad S_i \cap S_j \neq \emptyset$.
$u \neq v \Rightarrow \{u, v\} \in A$

Qu'est-ce-que peut dire de $S_1, S_2, S_3 \in G$.
$\forall u, v \in S_i \quad \{u, v\} \notin A$.

On a fait une réduction de 3 coloration à partition, mais le contraire est faux.

Les problem NP-complet se réduit l'un vers l'autre et vice versa.

Q9) Partition en clique à k-coloration :

$(G, 2)$ PART_cliques
(1)

G

Q10) $\quad 3\,col \leq$ PART_clique $\leq$ k-col.