# TD HAI602I Calculabilité/Complexité

## Année 2021-22

## Version 1.3

Université de Montpellier
Place Eugène Bataillon
34095 Montpellier Cedex 5

Rodolphe Giroudeau et Jean-Claude König
161, rue Ada
34392 MONTPELLIER Cedex 5
TEL : 04-67-41-85-40
Mail : {rgirou, konig}@lirmm.fr

Université de Montpellier
Licence HAI602I

18 janvier 2022
Durée: 30h

**C**alculabilité/Complexité
**T**D – Séance $n^o$ 1

---

# 1 Calculabilité

## 1.1 Divers

**Exercice 1 – Paradoxe**

Montrer que les problèmes suivants engendrent un paradoxe

1. Le conseil municipal d'un village vote un arrêté municipal qui enjoint à son barbier (masculin) de raser tous les habitants masculins du village qui ne se rasent pas eux-mêmes et seulement ceux-ci.
2. Un crocodile s'empare d'un bébé et dit à la mère : « si tu devines ce que je vais faire, je te rends le bébé, sinon je le dévore. »
   En supposant que le crocodile tienne parole, que doit dire la mère pour que le crocodile rende l'enfant à sa mère ?
   Une réponse usuelle de la mère est : « Tu vas le dévorer ! »

**Exercice 2 – Une preuve incorrecte**

Nous considérons la fonction suivante donné par l'algorithme 1 :

**Algorithm 1** La fonction de Collatz

```
while n ≠ 1 do
  if n = 0  mod 2 then
    n := n/2
  else
    n := 3 × n + 1
  end if
end while
```

Actuellement nous ne savons pas si cette fonction termine $\forall n$.

Est-ce que vous êtes d'accord avec la preuve suivante ?

« Si le problème de l'arrêt était décidable il suffirait de l'appliquer à ce programme pour savoir si son exécution s'arrête. Or, on ne sait pas si son exécution s'arrête. D'où la contradiction »

## 1.2 Variations sur le codage

**Exercice 3 – Codage de couples d'entiers**

Soit $Rang : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ tel que $Rang(x,y) = \frac{(x+y)(x+y+1)}{2} + x$

1. Donner une version récursive de la fonction $Rang$.
2. Donner la fonction inverse.
3. Calculer $Rang(4,5)$. Donner le couple pour lequel la valeur du codage est 8.

**Exercice 4 – Codage de triplets**

Soit $c$ la fonction de codage pour les couples d'entiers vue en cours.

1. Soit $h$ la fonction de codage pour les triplets définie par $h(x,y,z) = c(c(x,y),z)$. Quel est le doublet codé par 67 ? Quel est le triplet codé par 67 ?
2. le couple $(z,t)$ succède au couple $(x,y)$ si $c(z,t) = c(x,y) + 1$. Ecrire la fonction successeur qui prend en paramètre un couple et retourne le couple successeur.

**Exercice 5 – Etude d'une équation fonctionnelle dans $\mathbb{N}$**

Soit $f$ une application de $\mathbb{N}$ dans $\mathbb{N}$ telle que : $\forall (m,n) \in \mathbb{N}, \mathrm{f}(\mathrm{m}^2 + \mathrm{n}^2) = \mathrm{f}(\mathrm{m})^2 + \mathrm{f}(\mathrm{n})^2$. Nous voulons montrer que $f$ est :

— l'application nulle, donnée par : $\forall n \in \mathbb{N}, \mathrm{f}(\mathrm{n}) = 0$,

— l'application identité, donnée par $\forall n \in \mathbb{N}, \mathrm{f}(\mathrm{n}) = \mathrm{n}$.

Nous supposerons que $a$ est l'entier naturel $f(1)$.

1. Montrer que $f(0) = 0$. En déduire que $\forall n \in \mathbb{N}$, on a $f(n^2) = f(n)^2$.
2. Montrer alors que $a^2 = a$, donc que $a$ est égal à 0 ou à 1. Pour répondre, il suffit de prouver que l'égalité $f(n) = an$ est vrai pour $n = 0$ et $n = 1$, est vraie $\forall n \in \mathbb{N}$.
3. Vérifier successivement les égalités $f(2) = 2a, f(4) = 4a$ et $f(5) = 5a$.
4. Utiliser les valeurs de $f(4)$ et de $f(5)$ pour montrer que $f(3) = 3a$.
5. Utiliser les valeurs de $f(1)$ et de $f(5)$ pour montrer que $f(7) = 7a$.
6. Montrer que $f(8) = 8a, f(9) = 9a, f(10) = 10a$ et $f(6) = 6a$.
7. Observer que

$$\forall m, \text{ on a } \begin{cases} (2m)^2 + (m-5)^2 = (2m-4)^2 + (m-3)^2 \\ (2m+1)^2 + (m-2)^2 = (2m-1)^2 + (m+2)^2 \end{cases}$$

   Montrer que $\forall n$, on a $f(n) = an$.
8. Conclusion.

**Exercice 6 – Codage rationnels**

Proposer un codage pour les nombres rationnels.

**Exercice 7 – Codage des listes entiers**

Pour coder les listes d'entiers peut-on :

1. Faire la somme des entiers de la liste, et à somme égale prendre l'ordre lexicographique ?
2. Faire comme pour les mots : prendre les listes les plus courtes d'abord et à égalité de longueur l'ordre lexicographique ?

**Exercice 8 – Codage de listes d'entiers**

On ordonne les listes de la façon suivante :

$$\sigma(l) = \text{ somme des entiers de la liste} + \text{longueur de la liste}$$

Puis à valeur de $\sigma$ égale on ordonne dans l'ordre lexicographique.

On note $U_k$ l'ensemble des liste $l$ telles que $\sigma(l) = k$ et $u_k = |U_k|$.

1. Donner les ensembles $U_i, i = 0, \ldots, 4$.
2. Montrer que $u_k = 2^k, \forall k \geq 1$.
3. Quelle est la première liste de $U_k, \forall k \in \mathbb{N}^*$ et la dernière ?
4. Donner la fonction de codage en version itérative et récursive (resp. décodage).

**Exercice 9 – Codage d'entiers**

Soit la fonction $f$ suivante de $\mathbb{N}^* \to \mathbb{N}$ :

$$\begin{array}{rcl} f(n) & = & k \text{ si } n = 2^k \\ f(n) & = & f(n/2) \text{ si } n \text{ pair et n'est pas une puissance de 2} \\ f(n) & = & f((3n+1) \text{ sinon} \end{array}$$

Nous appelons $A_i = \{x | f(x) = i\}$.

1. Donner quelques éléments de $A_i, \forall i \in \{1, 2, 3, 4, 5, 6\}$.
2. Donner un algorithme qui prend $i$ en paramètre et qui affiche tous les éléments de $A_i$.
3. Donner un algorithme qui affiche $A_1 \cup A_2$.
4. Donner un algorithme qui affiche $A_4 \cup A_6$.

## 1.3 Diagonalisation

**Exercice 10 – Diagonalisation**

1. Montrer que l'ensemble des parties d'un ensemble $E$ infini dénombrable n'est pas dénombrable.
2. Que peut-on en conclure sur la cardinalité de l'ensemble des fonctions ? et de l'ensemble des programmes ?
3. Préciser le cas où $E$ est un ensemble fini (donc dénombrable),

**Exercice 11 – Diagonalisation**

1. Montrer que l'ensemble des sous-ensembles d'un ensemble dénombrable n'est pas dénombrable. Pour cela considérer un ensemble dénombrable $A = \{a_0, a_1, a_2, \ldots, \}$, et $S$ l'ensemble de ses sous-ensembles.

**Exercice 12 – Diagonalisation**

1. Soit une suite quelconque d'ensembles $E_i \subset \mathbb{N}$. Construire un ensemble qui n'appartient pas à cette suite (en vous inspirant de la diagonalisation).
2. Que pouvons-nous conclure sur l'ensemble des sous-ensembles de $\mathbb{N}$ ?

**Exercice 13 – Diagonalisation**

Monter que $[0, 1[$ n'est pas dénombrable. Conclusion

**Exercice 14 – Diagonalisation**

On considère l'ensemble $U$ des suites $(u_n)_{n\in\mathbb{N}}$ à la valeurs dans $\{0, 1\}$,c'est à dire $\forall n \in \mathbb{N}$ ; montrer que $U$ n'est pas dénombrable.

## 1.4 Dénombrabilité

**Exercice 15 – Ensemble fini/infini**

Un ensemble est fini si on ne peut pas le mettre en bijection avec une partie stricte de lui-même. Il est infini sinon.

Montrer que l'ensemble des entiers est infini.

**Exercice 16 – Taille des ensembles**

Soit $E$ un ensemble, et soit $\mathcal{P}(E)$ l'ensemble des parties de $E$. On a $|E| < |\mathcal{P}(E)|$.

Pour montrer ceci, on suppose qu'il existe une bijection de $E$ dans $P(E)$.

**Exercice 17 – Dénombrabilité**

1. Donner les bijections :
   (a) de $\mathbb{N}$ sur $\mathbb{N} - \{0\}$.
   (b) de $\mathbb{N}$ sur $2\mathbb{N}$.
   (c) de $\mathbb{N}$ sur $\mathbb{Z}$.
2. Est-ce que la fonction $f(n) = (-1)^n\lceil \frac{n}{2} \rceil$ est une bijection de $\mathbb{N}$ sur $\mathbb{Z}$ ?
3. Montrer que tout sous-ensemble $X \subset \mathbb{N}$ est dénombrable.
4. Il existe une application $f : X \to \mathbb{N}$ qui est injective si et seulement si $X$ est dénombrable.
5. Un produit fini d'ensembles dénombrables est dénombrable.
6. Il existe une application $f : \mathbb{N} \to \mathrm{X}$ qui est surjective si et seulement si $X$ est dénombrable.
7. Soit $E$ un ensemble dénombrable infini. Alors il existe une bijection de $\mathbb{N}$ sur $E$. Autrement dit, on peut numéroter les éléments de $E$, i.e. écrire $E = \{e_0, e_1, \ldots, e_n, \ldots\}$.
8. Montrer que $\mathbb{Q}$ est dénombrable.
9. Soit $(E_n)_{n\in\mathbb{N}}$ une famille dénombrable de sous-ensembles dénombrables d'un ensemble $E$. Montrer que la réunion $\cup_{n\in\mathbb{N}} E_n$ est dénombrable.
10. Soit $A = \mathbb{Q} \cap [0, 1]$ et $B = \mathbb{Q} \cap ]0, 1[$. Existe-t'il une bijection de $A$ vers $B$ ?

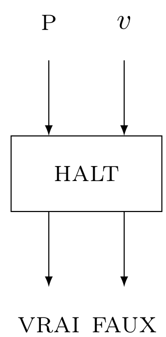


FIGURE 1 – Illustration graphique du programme HALT. P est un programme et $v$ des données.

## 1.5 Fonctions (non)-calculables

**Exercice 18 – Calculabilité**

Soit $f : \mathbb{N} \to \{0, 1\}$ une fonction totale non calculable.

1. Rappeler la définition d'une fonction totale et d'une fonction non calculable.
2. Construire une fonction $g : \mathbb{N} \to \mathbb{N}$ totale, croissante et non calculable à partir de $f$.

**Exercice 19 – Calculabilité**

Montrer que l'inverse d'une fonction $f$ calculable et bijective est calculable.

**Exercice 20 – Calculabilité**

Montrer qu'une fonction $f$ totale $\mathbb{N} \to \mathbb{N}$ est calculable si et seulement si son graphe

$$G = \{(x, f(x)|x \in \mathbb{N}\}$$

est décidable.

**Exercice 21 – Calculabilité**

Soient $E$ un ensemble et $\phi$ une fonction telle que $\phi(n)$ est égale au nombre d'éléments de $E$ strictement inférieur à $n$.

1. Montrer que $\phi$ est calculable si et seulement si $E$ est décidable.

## 1.6 Problèmes indécidables

**Exercice 22 – Variantes du problème l'arrêt**

1. SELF-HALT : le programme SELF-HALT($p$) s'arrête sur $p$ si HALT($p, p$) s'arrête où HALT($p, v$) désigne le problème de l'arrêt pour un programme $p$ appliqué à des données $v$.
2. ANTI-SELF-HALT : le programme ANTI-SELF-HALT($p$) s'arrête si et seulement si $p$ ne s'arrête pas.
3. Monter que les problèmes suivants sont indécidables.

(a) $\text{HALT}_\exists$ : le problème de l'arrêt existentiel, existe-t-il une entrée pour laquelle le programme s'arrête ?

(b) $\text{HALT}_\forall$ : le problème de l'arrêt universel, le programme s'arrête-t-il pour toutes les entrées ? La réduction est l même que la précédente.

(c) NEGVAL : le problème du test de valeur négative, la variable $v$ du programme prend-elle une valeur négative au cours du calcul ? Ce problème est à rapprocher de « cet indice de tableau évolue-t-il toujours dans les bornes du tableau ? »

(d) EQUIV : problème du test d'équivalence de programmes, les programmes $\text{P}_1$ et $\text{P}_2$ ont-ils le même comportement pour toutes les entrées ?

Pour illustrer l'intérêt de ce problème : on peut se poser la question concernant d'un programme source et d'un programme objet correspondant produit par un compilateur. Certaines optimisations tendantes changent le comportement du programme.

(e) $\text{RETURN}_0$ : problème du test de rendu nul, existe-t-il une entrée pour laquelle le programme retourne la valeur 0.

## 1.7 Théorème de Rice

**Exercice 23 – Calculabilité**

En vous inspirant du théorème de Rice, donnez le prédicat (indécidable) et la fonction contradictoire qui prouve par l'absurde le résultat d'indécidabilité pour chacun des exemples suivants : on ne peut décider si une procédure calcule

1. une fonction totale
2. une fonction injective
3. une fonction croissante
4. une fonction à valeurs bornées

## 1.8 Décidabilité et récursivement énumérable

**Exercice 24 – Récursivement énumérable**

Soit $A$ un ensemble énumérable et $P : A \to Bool$ un programme total. Alors l'ensemble $B := \{a \in A | P(a) = Vrai\}$ est énumérable.

**Exercice 25 – Calculabilité**

Soit $E$ l'ensemble $val(f)$ où $f$ est calculable partielle.

1. Montrer que $E$ est récursivement énumérable (inspirez-vous du fait que l'arrêt en $t$ unités de temps est décidable)

**Exercice 26 – Calculabilité**

Soit $f$ une fonction calculable, un ensemble $B$ et son image réciproque par $f$, $A$ :

$$A = f^{-1}(B) = \{x | f(x) \in B\}$$

1. Rappeler la définition d'un ensemble décidable et d'un ensemble récursivement énumérable.

2. A-t'-on $B$ décidable implique $A$ décidable ?
3. A-t'-on $B$ récursivement énumérable implique $A$ récursivement énumérable ?

**Exercice 27 – Calculabilité**

1. Montrer qu'un ensemble énuméré par une fonction calculable strictement croissante $f$ est décidable.
2. En déduire que tout ensemble récursivement énumérable non décidable contient un sous-ensemble infini et décidable.

**Exercice 28 – Calculabilité**

1. Montrer que tout ensemble récursivement énumérable peut-être énuméré par une fonction sans répétition.

**Exercice 29 – Calculabilité**

Soient $A$ et $B$ deux ensembles décidables :

1. Est-on sûr que le complémentaire de $A$ est décidable ?
2. Est-on sûr que l'union de $A$ et $B$ est décidable ?
3. Est-on sûr que l'intersection de $A$ et $B$ est décidable ?
4. Même question en remplaçant décidables par récursivement énumérables.

**Exercice 30 – Calculabilité**

1. soit $A$ un ensemble décidable de couples d'entiers. Montrer que la projection de $A$ à savoir $E = \{x | \exists y \text{ tel que } (x, y) \in A\}$ est récursivement énumérable.
2. Montrer que réciproquement tout ensemble récursivement énumérable est la projection d'un ensemble décidable.

**Exercice 31 – Concept de la réduction**

Pour deux sous-ensembles $A$ et $B$ de $\mathbb{N}$, on dit que $A$ se réduit à $B$ (ce qu'on note $A \propto B$) si il existe une fonction totale (récursive) totale $f$ telle que pour tout $x \in \mathbb{N}$, $x \in A$ ssi $f(x) \in B$.

1. Montrer que si $B$ est décidable et $A \propto B$, alors $A$ est décidable.
2. Montrer que si $B$ est récursivement énumérable et $A \propto B$, alors $A$ est récursivement énumérable .

## 1.9 Sur le point fixe

**Exercice 32 – Exemples**

Donner les points fixes pour les fonctions suivantes :

1. $\mathbb{N} \to \mathbb{N}, \mathrm{x} \mapsto \mathrm{x} + 1$
2. $\mathbb{N} \to \mathbb{N}, \mathrm{x} \mapsto 0 * \mathrm{x}$
3. $$\mathbb{N} \to \mathbb{N}, \mathrm{x} \mapsto \begin{cases} x/2 \text{ si } x \text{ est pair} \\ x \end{cases}$$

4.
$$\mathbb{N} \to \mathbb{N}, \mathrm{x} \mapsto \begin{cases} x \text{ si } x \geq n \\ x+1 \end{cases}$$

5.
$$\mathbb{N} \to \mathbb{N}, \mathrm{x} \mapsto \begin{cases} x \text{ si } x \geq n \\ x-1 \end{cases}$$

6.
$$\mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}, \langle \mathrm{x}, \mathrm{y} \rangle \mapsto \begin{cases} \langle x-1, y+1 \rangle \text{ si } x > 0 \\ \langle x, y \rangle \end{cases}$$

# 2 Complexité

## 2.1 Rappel

**Exercice 33 – Une certaine idée de la complexité**

Soit la fonction $C$ suivante :

```
int pf(int x)
    {
    int y=pg(x)
    return ph(y)
    }
```

1. Quelle est la complexité du calcul de $pf$ si $pg$ est de complexité $O(n^4)$, $ph$ de complexité linéaire et si $g(n) < n^2$ ($g$ étant la fonction calculée par $pg$) ?
2. Si $ph$ s'exécute en temps polynomial, à quelle condition le calcul de $pf$ se fait-il en temps polynomial ?
3. Si les hypothèses de la question précédente est vérifiée que peut-on en déduire si la fonction $h$ calculée par $ph$ se calcule en temps polynomial ?
4. Soit le calcul de Fibonacci en utilisant directement la formule de récurrence :$f_0 = 1, f_1 = 1, n > 1, f_n = f_{n-1} + f_{n-2}$. Montrons que le nombre d'additions nécessaires pour faire le calcul est compris entre $\sqrt{2}^n$ et $2^n$ ?
5. Comment améliorer pour que ce nombre soit en $O(n)$ ? Peut-on en déduire qu'il existe un algorithme qui calcule $f_n$ avec un nombre d'additions polynomial par rapport à la taille de la donnée ? Pourquoi ?
6. Questions difficiles : comment calculer $f_n$ avec un nombre d'additions et de multiplications polynomial par rapport à la taille de la donnée ? Peut-on trouver un algorithme qui s'exécute en temps polynomial par rapport à la taille de la donnée ?

**Exercice 34 – Sur le codage d'un graphe**

Donner la taille en nombre de bits pour coder un graphe en utilisant une matrice d'adjacence et liste chaînées.

**Exercice 35 – Certificat**

Si pour un problème $\Pi$ vous avez un certificat polynomial pour une réponse positive et un certificat polynomial pour une réponse négative. Que pouvez-vous conclure ? Justifiez votre réponse

On connait un algorithme simple en $O(\sqrt{n})$ pour savoir si un nombre $n$ est premier. Peut-on en déduire que savoir si un nombre est premier.

On peut savoir si $n$ peut s'écrire comme le produit de deux nombres premiers et on connait un algorithme en $O(\sqrt{n})$. Peut-on en déduire que ce problème est dans $P$ ? Quel serait l'impact si ce problème était dans $P$ ?

**Exercice 36 – Puissance de calcul**

Tous les 4 ans la puissance des machines est multipliée environ par 8. Vous avez deux algorithmes $A$ et $B$ l'un dont le temps d'exécution est proportionnel à $n^3$ et l'autre dont le temps d'exécution est proportionnel à $2^n$. Avec les deux algorithmes vous traitiez un problème de taille $n = 10$ en $1s$, il y a 40 ans. Quelle est la taille des problèmes que vous êtes capables de traiter aujourd'hui avec chacun des deux algorithmes en $1s$ ?.

## 2.2 Autour des classes $\mathcal{P}$ et $\mathcal{NP}$

**Exercice 37 – 2-Satisfaisabilité**

1. Montrer en calculant le nombre de clauses créées et le nombre de variables ajoutées que la réduction de Satisfaisabilité à 3-Satisfaisabilité vue en cours est bien polynomiale.
2. Sachant que 2-Satisfaisabilité peut-être résolu en temps polynomial. Appliquer l'algorithme pour les deux instances suivantes :
   — $\phi = (x_1 \vee x_2) \wedge (x_3 \vee \bar{x}_4) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_3) \wedge (x_4 \vee \bar{x}_2)$.
   — $\phi = (x_1 \vee x_3) \wedge (x_2 \vee \bar{x}_4) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_3) \wedge (x_4 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3)$.
3. Quel problème d'optimisation pouvons-nous étudier dans le cas où la réponse est négative à l'existence d'une solution pour 2-Satisfaisabilité ?

**Exercice 38 – Algorithme non-déterministe pour le problème de 3-coloration**

Proposer un algorithme linéaire ($O(n)$) pour le problème 3-coloration.

**Exercice 39 – Classification dans $\mathcal{NP}$ ou dans $\mathcal{P}$**

Problème P1
**Entrée :** $G = (V, E)$ un graphe non orienté.
**Question :** Existe t'il un cycle de longueur égale à $\lfloor \frac{|V|}{2} \rfloor$ ?

Problème P2
**Entrée :** $G = (V, E)$ un graphe non orienté.
**Question :** Existe t'il un cycle de longueur égale à 4 ?

Problème P3
**Entrée :** $G = (V, E)$ un graphe non orienté.
**Question :** Existe t'il un simple chemin entre $u$et $v$ de longueur inférieure ou égale à $k$ ?

Problème P4
**Entrée :** $G = (V, E)$ un graphe non orienté et un entier $k$.
**Question :** Existe t'il un arbre couvrant tous les sommets de $G$ ayant mois de $k$ feuilles ?

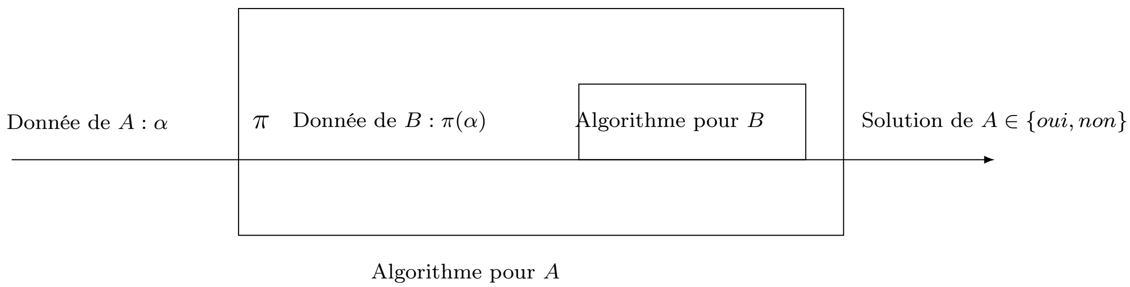


Figure 2 – Réduction de Karp.

1. Classer les problèmes en fonction de $\mathcal{P}$ et $\mathcal{NP}$.

## 2.3 Réduction polynomiale

**Exercice 40 – Réduction de Karp : une vision algorithmique**

- Complexité en temps de la réduction en $O(|\alpha|^{c_1}) \Rightarrow |\pi'(\alpha)| \in O(|\alpha|^{c_1})$,
- Supposons $B \in P$ : algorithme pour une donnée $\beta$ de en $O(|\beta|^{c_2})$.
- Algorithme pour $A$ admet une complexité de ?
- Contradiction, donc $A$ est NP-difficile implique $B$ est NP-difficile.

1. Compléter les éléments précédents.

**Exercice 41 – Concept de la réduction (suite)**

Pour réduire un problème $A$ à un problème $B$, il suffit de montrer que la résolution de $B$ permet de résoudre $A$ à condition qu'une solution à $B$ soit disponible.

Pour illustrer, supposons que $A$ est le problème suivant : $A(n) =$ le plus petit nombre premier qui est plus grand $n$, et $B$ le problème de décision $B = \{n|n \text{ is prime }\}$.

1. Donner pour quelques valeurs de $n$ la valeur de $A(n)$.
2. Proposer une réduction du problème $A$ au problème $B$.

**Exercice 42 – Réduction**

Montrer que les deux problèmes Problème du carré d'un entier et Problème de la multiplication se réduisent l'un à l'autre.

L'addition, la soustraction et la division soient des opérations autorisées.

Problème de la multiplication
**Entrée :** Soient $a \in \mathbb{N}$ et $b \in \mathbb{N}$.
**Question :** Peut-on multiplier $a$ et $b$ ?

Problème du carré d'un entier
**Entrée :** Soit $a \in \mathbb{N}$.
**Question :** Peut-on élevé au carré $a$ ?

**Exercice 43 – Réduction entre deux problèmes polynomiaux**

2-Satisfaisabilité

**Entrée :** Etant donnée une formule conjonctive $\phi$ sur $n$ variables et $m$ clauses chacune de taille deux.

**Question :** Existe-t-'il une affectation de valeurs de vérité aux variables qui satisfasse $\phi$ ?

2-coloration

**Entrée :** Soit $G = (V, E)$, et deux couleurs.

**Question :** Existe-t'il une 2-coloration valide, *i.e.* une fonction totale $f : \mathbb{N} \to \{1, 2\}$ tel $f(u) \neq f(v), \forall (u, v) \in E$ ?

1. Montrer qu'il existe une réduction polynomial entre 2-coloration et 2-Satisfaisabilité.
2. Conclure sur la complexité du problème 2-Satisfaisabilité.

**Exercice 44 – Problèmes équivalent polynomialement**

Pour les problèmes suivants, indiquez si les problèmes sont polynomialement équivalents.

1. Arbre couvrant de poids minimum et Arbre couvrant de poids maximum

   Arbre couvrant de poids minimum (Minimum spanning tree)
   **Entrée :** Un graphe $G = (V, E)$
   **But :** Trouver un sous-graphe connexe de poids minimum

   Arbre couvrant de poids maximum (Maximum spanning tree)
   **Entrée :** Un graphe $G = (V, E)$
   **But :** Trouver un sous-graphe connexe de poids maximum

2. Arborescence des plus courts chemins et Plus long Chemin

   Arborescence des plus courts chemins (Shortest Path)
   **Entrée :** $G = (V, E)$ un graphe orienté
   **But :** Trouver une arborescence des plus court chemins

   Un plus long chemin (Longuest Path)
   **Entrée :** $G = (V, E)$
   **But :** Trouver un plus long chemin sans répétition de sommets

3. Coupe de valeur maximale et Coupe de valeur minimale

   Coupe de valeur maximale (Maximum Cut)
   **Entrée :** $G = (V, E)$ un graphe orienté
   **But :** Trouver une coupe de valeur maximale

   Coupe de valeur minimale (Minimum Cut)
   **Entrée :** $G = (V, E)$
   **But :** Trouver une coupe de valeur minimale

4. Couplage maximum de valeur maximum et Couplage minimum de valeur minimum

    Couplage maximum de valeur maximum (Maximum Cut)
    **Entrée :** $G = (V, E)$ un graphe, et une valuation sur les arêtes
    **But :** Trouver un couplage maximum de poids maximum

    Couplage minimum de valeur minimum (Minimum Cut)
    **Entrée :** $G = (V, E)$ un graphe, et une valuation sur les arêtes
    **But :** Trouver un couplage maximum de poids minimum

5. Voyageur de Commerce de coût minimum et Voyageur de Commerce de coût maximum

    Voyageur de Commerce de coût minimum (Maximum TSP)
    **Entrée :** $G = (V, E)$ un graphe complet, et une valuation sur les arêtes
    **But :** Trouver un chemin Hamiltonien de poids minimum

    Voyageur de Commerce de coût maximum (Minimum TSP)
    **Entrée :** $G = (V, E)$ un graphe complet, et une valuation sur les arêtes
    **But :** Trouver un chemin Hamiltonien de poids maximum

**Exercice 45 – Problème de décision**

Mettre les problèmes suivants sous forme de problème de décision et évaluer la taille de leurs instances.

1. de savoir s'il existe un chemin entre deux sommets disjoints dans un graphe ;
2. de connaitre la distance entre deux sommets disjoints dans un graphe ;
3. de connaitre la longueur de la chaîne maximum dans un graphe pondéré.

**Exercice 46 – Optimisation versus décision**

Soit le problème du stable de taille $k$ :

Stable (Stable)
**Entrée :** Un graphe non orienté $G = (X, E)$
**Question :** Existe-t'il un stable (c'est à dire un sous-ensemble de sommets tel que deux sommets de ce sous-ensemble ne soient jamais reliés par une arête) de taille $k$

et sa version optimisation

Max Stable (MaxStable)
**Entrée :** Un graphe non orienté $G = (X, E)$
**Question :** Trouver un stable (c'est à dire un sous-ensemble de sommets tel que deux sommets de ce sous-ensemble ne soient jamais reliés par une arête) de taille maximum

1. Montrer que S'il existe un algorithme polynomial qui résout le problème de stabilité maximum alors la version décisionnelle est résoluble, lui aussi, en temps polynomial.

2. Montrer que s'il existe un algorithme qui résout le problème de stable de taille $k$ en temps polynomial alors le problème de stabilité maximum est résoluble, lui aussi, en temps polynomial.

## 2.4 Autour des classes $\mathcal{NP}$ et $\mathcal{NP}$-complet

### 2.4.1 Certificat

**Exercice 47 – Certificats polynomiaux et réductions polynomiales**

1. Quels sont les certificats des problèmes de décision suivants et sont-ils des certificats polynomiaux :
   (a) 2-Partition.
   (b) Circuit Hamiltonien.
   (c) Satisfaisabilité.
   (d) Clique.
   (e) 3-coloration

   Mêmes questions pour les co-problèmes (la question est formulée de façon négative).

**Exercice 48 – Certificat positif**

Pour les problèmes suivants, donner le certificat positif.

1. 3-Satisfaisabilité .
2. k-Coloration.
3. Sous-somme Maximale.
4. 2-Partition.
5. Circuit Hamiltonien.
6. Satisfaisabilité.
7. Clique.
8. 3-coloration :
9. Soit le problème CHEMIN=$\{< G, s, t, > |G$ est un graphe orienté possédant un chemin de $s$ à $t\}$

## 2.5 Propriétés des classes $\mathcal{NP}$ et $\mathcal{NP}$-complet

**Exercice 49 – Propriétés des classes $\mathcal{NP}$ et $\mathcal{NP}$-complet**

Soient $A$ et $B$ deux langages.

Prover ou réfuter les deux assertions suivantes :

1. Si $A$ et $B$ sont dans $\mathcal{NP}$, alors on a $A \cup B \in \mathcal{NP}$ et $A \cap B \in \mathcal{NP}$.
2. Si $A$ et $B$ sont $\mathcal{NP}$-complet, alors ni $A \cup B$ et ni $A \cap B$ peuvent être $\mathcal{NP}$-complet.

### 2.6 Classes $co\mathcal{NP}$ et $co\mathcal{NP}$-complet

**Exercice 50 – Définition de $co\mathcal{NP}$ par les langages formels**

Un langage $A$ est dans $co\mathcal{NP}$ si et seulement si il existe un polynôme $p(n)$ et un langage $B \in \mathcal{P}$ tels que

$$x \in A \Leftrightarrow \forall y \in \{0,1\}^{p(|x|)} (x, y) \in B$$

**Exercice 51 – Propriétés pour $co\mathcal{NP}$**

Supposons que $A \in \mathcal{NP}$ et que $B \in co\mathcal{NP}$. Nous supposerons pour cet exercice $\mathcal{NP} \neq co\mathcal{NP}$. Montrer la véracité ou trouver un contre-exemple aux assertions suivantes

1. $\bar{A} \cup B \in co\mathcal{NP}$.
2. $A \cap \bar{B} \in co\mathcal{NP}$.
3. $A \cup B \in co\mathcal{NP}$.
4. $A \cap B \in co\mathcal{NP}$.

**Exercice 52 – Absence de certificat positif**

Considérons le problème suivant :

Co-Voyageur de Commerce (COTSP)

**Entrée :** Un ensemble de $m$ villes $X$, un ensemble de routes entre les villes $E$. Une fonction de coût $v : E \to \mathbb{N}$ où $v(x, y)$ est le coût de déplacement de $x$ à $y$, $k \in \mathbb{N}$.

**Question :** Existe-il aucun cycle Hamiltonien de distance inférieure ou égale à $k$ ?

Est-ce que ce problème appartient à la classe $\mathcal{NP}$ ?

**Exercice 53 – Propriétés de $co\mathcal{NP}$**

1. Montrer que si $\pi$ est un problème $\mathcal{NP}$-complet tel que $\bar{\pi} \in \mathcal{NP}$ alors nous obtenons $co\mathcal{NP}$=$\mathcal{NP}$.
2. Montrer que si $co\mathcal{NP} \neq \mathcal{NP}$ alors $\mathcal{P} \neq \mathcal{NP}$.
3. Est-ce que nous pouvons avoir $\mathcal{P} \neq \mathcal{NP}$ et $co\mathcal{NP}$=$\mathcal{NP}$ ?

**Exercice 54 – Problème $co\mathcal{NP}$-complet**

De tout problème dans $\mathcal{NP}$, on peut construire un problème dual dans $co\mathcal{NP}$de problèmes suivants

1. Satisfaisabilité (SAT)
   **Entrée :** Etant donné une formule booléenne
   **Question :** Existe-t-il une assignation de ses variables qui la rend vraie ?
2. Chemin Hamiltonien (HC)
   **Entrée :** $G = (V, E)$
   **Question :** Existe-t'il un chemin Hamiltonien ?
3. Clique (Clique)
   **Entrée :** $G = (V, E)$, $k \in \mathbb{N}^*$
   **Question :** Existe-t'il une clique de taille $k$ ?

**Exercice 55 –** *co$\mathcal{NP}\cap\mathcal{NP}$*

Considérons le problème PROBLÈME DU FLOT MAXIMUM.

PROBLÈME DU FLOT MAXIMUM
**Entrée :** $G=(V,E,c)$ une source $s$ et $t$, et $K\in\mathbb{N}$.
**Question :** Est-il vrai que $G$ possède un flot de valeur au moins $K$ entre $s$ et $t$ ?

Montrer que PROBLÈME DU FLOT MAXIMUM $\in\mathcal{NP}\cap co\mathcal{NP}$.

**Exercice 56 –** *co$\mathcal{NP}\cap\mathcal{NP}$*

Considérons le problème PROBLÈME DU COUPLAGE PARFAIT.

PROBLÈME DU COUPLAGE PARFAIT
**Entrée :** $G=(U\cup V,E)$.
**Question :** Est-il vrai que $G$ possède un couplage parfait ?

Montrer que PROBLÈME DU COUPLAGE PARFAIT $\in\mathcal{NP}\cap co\mathcal{NP}$.

## 2.7 Classe $\mathcal{NP}$-complétude

### 2.7.1 Autour de la Satisfaisabilité

**Exercice 57 – Satisfaisabilité optimale**

Nous considérons des problèmes de satisfaisabilité maximum o nous cherchons une affectation de valeurs de vérité sur les variables d'une formule conjonctive $\phi$ qui vise à satisfaire non pas toutes les clauses mais un nombre maximum d'entre elles.

La variante décisionnelle de SATISFAISABILITÉ MAXIMALE est définie comme suit :

SATISFAISABILITÉ MAXIMALE (MaxSAT)
**Entrée :** Etant donné une formule conjonctive $\phi$ sur $n$ variables et $m$ clauses et un entier $k\leq m$,
**Question :** existe-t-'il une affectation de valeurs de vérité aux variables de $\phi$ qui satisfait au moins $k$ clauses ?

1. Montrer SATISFAISABILITÉ $\propto$ SATISFAISABILITÉ MAXIMALE. En déduire une preuve pour K-SATISFAISABILITÉ $\propto$ K-SATISFAISABILITÉ MAXIMALE.
2. Nous voulons prouver maintenant que 2-SATISFAISABILITÉ MAXIMALE est $\mathcal{NP}-$difficile.
   (a) Peut-on procéder à la réduction 2-SATISFAISABILITÉ $\propto$ 2-SATISFAISABILITÉ MAXIMALE ?
   (b) Nous allons procéder à la réduction suivante :
   3-SATISFAISABILITÉ MAXIMALE $\propto$ 2-SATISFAISABILITÉ MAXIMALE.
   Considérons une instance de 3-SATISFAISABILITÉ MAXIMALE, sur les variables $x_1,x_2,\ldots,x_n$ et les clauses $C_1,\ldots,C_m$ avec $|C_i|=3,\forall i$. On lui associe $\phi'$ définie sur les variables $x_1,x_2,\ldots,x_n,y_1,\ldots,y_m$ ($m$ nouvelles variables) et $10m$ clauses $C'_1,\ldots,C'_{10m}$ construites comme suit : considérons la clause $C_i=a_i\vee b_i\vee d_i$ où $a_i,b_i$ et $d_i$ sont des littéraux de variables $x_1,x_2,\ldots,x_n$ à $C_i$, nous lui associons l'ensemble de 10 nouvelles clauses suivantes :

$$\{a_i, b_i, d_i, (\bar{a}_i \vee \bar{b}_i), (\bar{a}_i \vee \bar{d}_i), (\bar{b}_i \vee \bar{d}_i), (a_i \vee \bar{y}_i), (b_i \vee \bar{y}_i), (d_i \vee \bar{y}_i), y_i\}$$

La formule $\phi'$ est obtenue par la conjonction de $10m$ clauses ainsi construites. Montrer que 2-Satisfaisabilité Maximale est $\mathcal{NP}-$difficile.

3. La variante décisionnelle de la version générale de Satisfaisabilité Minimale est définie comme suit :

   Etant donné une formule conjonctive $\phi$ sur $n$ variables et $m$ clauses et un entier $k \leq m$, existe-t-'il une affectation de valeurs de vérité aux variables de $\phi$ qui satisfait au plus $k$ clauses ?

   Montrer que Satisfaisabilité Minimale est $\mathcal{NP}$-complet à partir de Satisfaisabilité Maximale. Pour cela aidez-vous de la réduction polynomiale suivante :

   Considérons une instance de 2-Satisfaisabilité Maximale, notée $(\phi, K_{max})$, où $\phi$ est une formule conjonctive défini sur les variables $x_1, x_2, \ldots, x_n$ et les clauses $C_1, \ldots, C_m$. On lui associe $\phi'$ définie sur les variables $x_1, x_2, \ldots, x_n, y_1, \ldots, y_m$ ($m$ nouvelles variables) et $2m$ clauses $C'_1, \ldots, C'_{2m}$ construites comme suit : considérons la clause $C_i = a_i \vee b_i$ où $a_i, b_i$ sont des littéraux de variables $x_1, x_2, \ldots, x_n$ à $C_i$, nous lui associons l'ensemble de 2 nouvelles clauses suivantes :

$$(\bar{a}_i \vee y_i), (\bar{b}_i \vee \bar{y}_i)$$

   La formule $\phi'$ est obtenue par la conjonction de $2m$ clauses ainsi construites. Nous posons $K_{min} = 2m - K_{max}$ Montrer que 2-Satisfaisabilité Minimale est $\mathcal{NP}-$difficile en montrant qu'au plus $K_{min}$ clauses sont satisfaite dans $\phi'$ si et seulement si au moins $K_{max}$ clauses sont satisfaites dans $\phi$..

**Exercice 58 – Autour de Satisfaisabilité**

Non égal Satisfaisabilité (NAESAT)

**Entrée :** Etant donnée une formule conjonctive $\phi$ sur $n$ variables et $m$ clauses

**Question :** Existe-t-'il une affectation de valeurs de vérité aux variables qui satisfasse $\phi$ tel que chaque clause à une littéral à vrai et un à faux ?

Montrer que Non égal Satisfaisabilité est $\mathcal{NP}$ – *complet*. La preuve se fera à partir de Satisfaisabilité

**Exercice 59 – Autour de Satisfaisabilité (suite)**

Coupe maximum (CUT)

**Entrée :** Soit $G = (V, E)$ un graphe non orienté, $k \in \mathbb{N}$

**Question :** Existe t'il une partition de sommets en deux sous-ensembles $V_1$ et $V_2$ tel que le nombre d'arêtes entre $V_1$ et $V_2$ est $k$ ?

Réduire Non égal 3-Satisfaisabilité à Coupe maximum. Conclure.

Remarque : vous verrez en master que coupure min est polynomiale même si les arêtes ont des poids.

### 2.7.2 Problèmes autour des graphes

**Exercice 60 – Problème de la coloration**

Montrer que 3-coloriable est $\mathcal{NP}$-complet (réduction à partir de 3-SATISFAISABILITÉ ).

**Exercice 61 – VOYAGEUR DE COMMERCE**

VOYAGEUR DE COMMERCE (TSP)

**Entrée :** Un ensemble de $m$ villes $X$, un ensemble de routes entre les villes $E$. Une fonction de coût $v : E \to \mathbb{N}$ où $v(x, y)$ est le coût de déplacement de $x$ à $y$, $k \in \mathbb{N}$.

**Question :** Existe-il un cycle Hamiltonien de distance inférieure ou égale à $k$ ?

Montrer que VOYAGEUR DE COMMERCE est $\mathcal{NP}$-complet. (La preuve se fait à partir de CYCLE HAMILTONIEN). Qu'en est t'il si on autorise l'inégalité triangulaire $\forall i, j, k, c_{ik} \leq c_{ij} + c_{jk}$ ?

**Exercice 62 – RECOUVREMENT DE SOMMETS**

On veut montrer que le problème RECOUVREMENT DE SOMMETS est $\mathcal{NP}$-complet. La preuve se fera à partir de 3-SATISFAISABILITÉ .

**Aide pour la transformation polynomiale :** Considérons les variables $x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n$ et $n$ arêtes $(x_i, \bar{x}_i), \forall i = 1, \ldots, n$. Nous considérons $m$ triangles constitués des littéraux. Pour une clause $C_i$ nous notons $c_{i_1}, c_{i_2}, c_{i_3}$ et nous relions le sommet $x_i$ à un sommet d'un triangle noté $C_{j_k}, k = 1, 2, 3$ si la variable $x_i$ apparaît dans la clause $C_j$ à la position $k$. littéraux

**Exercice 63 – RECOUVREMENT DE SOMMETS (suite)**

Montrer que le RECOUVREMENT DE SOMMETS reste $\mathcal{NP}$-complet même si tous les sommets sont de degrés pairs.

**Exercice 64 – Réductions autour de Hamiltonisme**

Montrer que les problèmes sont tous NP-complets si l'un d'eux l'est :

1. CYCLE HAMILTONIEN dans un graphe non orienté
2. CHAÎNE HAMILTONIENNE dans un graphe non orienté
3. CIRCUIT HAMILTONIEN dans un graphe orienté
4. CHEMIN HAMILTONIEN dans un graphe orienté
5. CYCLE HAMILTONIEN dans un graphe biparti non orienté
6. CHAÎNE HAMILTONIENNE dans un graphe biparti non orienté

**Exercice 65 – Quelques réductions classiques**

1. Montrer que RECOUVREMENT DE SOMMETS vertex cover est $\mathcal{NP}-$complet (Preuve à partir de 3-SATISFAISABILITÉ ).
2. Montrer que CLIQUE est $\mathcal{NP}$-complet (preuve à partir de 3-SATISFAISABILITÉ ).
3. Soit le problème

   2-PARTITION À VALEURS PAIRES (Partition à Valeurs Paires)

   **Entrée :** Un ensemble fini $A$ et une fonction $\tau$ de $A$ dans $\mathbb{N}$ telle que $\forall e \in A$ $\tau(e)$ est **pair**.

   **Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

   Prouver que le problème 2-PARTITION À VALEURS PAIRES est $\mathcal{NP}$-complet.

4. On rappelle l'énoncé du problème Sous-somme Maximale.

   Sous-somme Maximale (Sous-somme Maximale)
   **Entrée :** Un ensemble $E = \{e_1 \dots e_n\}$, une fonction $\tau :\ E \to \mathbb{N}$, et un entier $T$.
   **Question :** Existe-t-il un sous-ensemble $E' \subseteq E$ tel que $T = \sum_{e \in E'} \tau(e)$

   Prouver, à partir de *Partition* que *Somme de sous ensemble* est $\mathcal{NP}$-complet.
5. Le professeur Ykcid prétend avoir un algorithme polynomial pour construire, quand elle existe,une solution au problème *Somme de sous ensemble* pour tout ensemble quand le poids $\tau$ de chaque élément est **impair** (et ce quelle que soit la valeur de $T$).

   Prouvez lui que soit il a démontré que $\mathcal{P} = \mathcal{NP}$, soit il s'est trompé.

   *Indication* : on pourra considérer la valeur $S_E = \sum_{e \in E} \tau(e)$ [1], la valeur $S_{imp}$ qui sera le plus petit nombre impair strictement supérieur à $\sum_{e \in E} \tau(e)$, et une fonction $\tau' = 2\tau + S_{imp}$.

   Si un sous ensemble $E' \subseteq E$ est tel que $\sum_{e \in E'} \tau(e) = \frac{S_E}{2}$ et si $|E'| = k$, quelle est la valeur de $\sum_{e \in E'} \tau'(e)$ ?
6. Montrer que Partition en triangles est $\mathcal{NP}-$complet (preuve à partir de Exact cover by $3-SET$.

   Couverture Exacte d'ensemble (Couverture Exacte d'ensemble)
   **Entrée :** Un ensemble fini $X$ avec $|X| = 3q$ et une collection $C$ de sous-ensembles de trois éléments de $X$.
   **Question :** Est-ce que $C$ contient un couverture exacte de $X$, i.e. une sous-collection $C' \subseteq C$ tel que chaque élément de $X$ apparaît exactement une et une seule fois dans $C'$

   Partition en Triangles (Partition en Triangles)
   **Entrée :** Un graphe $G = (V, E)$ avec $|V| = 3q$, $q \in \mathbb{N}$.
   **Question :** Est-ce qu'il existe une partition de $V$ en $q$ ensembles disjoints $V_1, V_2, , \dots, V_q$ de trois sommets chaque, tel que pour chaque $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$ les trois arêtes appartiennent à $E$ ? Autrement dit peut-on couvrir le graphe par des triangles ou logiques de taille trois ?

**Exercice 66 – Quelques réductions classiques**

Nous supposerons que les problèmes suivants sont $\mathcal{NP}$-complets.

Circuit Hamiltonien (Circuit Hamiltonien)
**Entrée :** Un graphe $G = (V, E)$
**Question :** Existe t'il un cycle qui visite chaque sommet exactement une fois ?

Montrer que les problèmes suivants sont $\mathcal{NP}$-complets :

1. Recouvrement de sommets à partir de 3-Satisfaisabilité

   **Aide pour la transformation polynomiale :** Considérons les variables $x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_n, \bar{x}_n$ et $n$ arêtes $(x_i, \bar{x}_i), \forall i = 1, \dots, n$. Nous considérons $m$ triangles constitués des littéraux. Pour une clause $C_i$ nous notons $c_{i_1}, c_{i_2}, c_{i_3}$ et nous relions le sommet $x_j$ à un sommet d'un triangle noté $C_{j_k}, k = 1, 2, 3$ si la variable $x_j$ apparaît dans la clause $C_j$ à la position $k$. littéraux

1. on s'intéresse au cas $S_E$ pair sinon la réponse à *Partition* est *non* pour l'instance $E, \tau$

2. Voyageur de Commerce (La preuve se fait à partir de Cycle Hamiltonien). Qu'en est t'il si on autorise l'inégalité triangulaire ($\forall i, j, k, c_{ik} \leq c_{ij} + c_{jk}$ ?
3. Le problème Chemin Hamiltonien.

   Chemin Hamiltonien (Chemin Hamiltonien)
   **Entrée :** Un graphe orienté $G = (V, E)$
   **Question :** Existe t'il un cycle orienté Hamiltonien (Le chemin peut commencer à n'importe quel sommet pour finir à n'importe quel sommet) ?

4. Le problème du Plus long Chemin

   Plus long Chemin (Plus Long Chemin)
   **Entrée :** Un graphe orienté $G = (V, E)$, $L \in \mathbb{N}$
   **Question :** Existe t'il un chemin simple (sans boucle) de $s$ à $t$ avec au moins $L$ arcs ?

5. Le problème du Sac à Dos

   Sac à Dos (Sac à Dos)
   **Entrée :** n ensemble $X$ d'objets à valeurs dans $\mathbb{N}$ et, pour chaque objet $x$, un volume $c_x \in \mathbb{N}$ et une utilitté $a_x \in \mathbb{N}$ ; une capacité $b \in \mathbb{N}$
   **Question :** rouver un sous-ensemble d'objets $X'$ tel que $\sum_{x \in X'} c_x \leq b$ et qui maximise $\sum_{x \in X'} a_x$ ?

6. Le problème d'un Plus Court Chemin Sous Contrainte

   Plus Court Chemin Sous Contrainte (Plus Court Chemin Sous Contrainte)
   **Entrée :** Soit un graphe $G = (V, E)$, deux entiers $c, \tau \in \mathbb{N}$, une longueur d'arc $c_{ij}, \forall (i, j) \in E$ et un temps de traversé $\tau_{ij}, \forall (i, j) \in E$.
   **Question :** Existe t'il dans le graphe $G$ un chemin d'un sommet $s$ à un sommet $t$ dont le nombre de sommets est au plus $c$ et dont le temps de traversé est d'au plus $\tau$ ?

   **Idée** : Utiliser le problème du Sac à Dos.
7. Le problème de Steiner
   **Définition :** Soit $G$ un graphe et soit $T \subseteq V$. Un arbre de Steiner pour $T$ dans $G$ est un arbre $S$ avec $T \subseteq S \subseteq V$ et $E(S) \subseteq E$. Les sommets de $T$ seront appelés les sommets terminaux et les autres de $S \backslash T$ les sommets de l'arbre de Steiner.

   Steiner (Steiner)
   **Entrée :** Soit un graphe $G = (V, E, c)$, $c \in \mathbb{N}$ et soit $T \subseteq V$
   **Question :** Existe t'il un arbre de Steiner $S$ pour $T$ dont le poids soit minimum ?

   (a) Que retrouvez-vous dans le cas où $T = \{i, j\}, i \neq j$ ?
   (b) De même pour $T = \{1, 2, \ldots, n\}$ ?
   (c) Montrer que le problème de Steiner est $\mathcal{NP}$-complet.

8. Montrer que les problèmes suivants sont $\mathcal{NP}-$complets (Utiliser le problème Chemin Hamiltonien. Soit un graphe $G = (V, E)$, un ensemble $L \subset V$, un entier $k$, existe-t'-il un arbre couvrant $T$ de $G$ tel que :
   (a) L'ensemble des feuilles de $T$ est $L$ ?
   (b) Il n'existe pas de feuilles de $T$ à l'extérieur de $L$ ?
   (c) $T$ admet $k$ feuilles ?
   (d) $T$ admet au plus $k$ feuilles ?
   (e) $T$ admet au moins $k$ feuilles ? (Utiliser le problème Ensemble Domimant Connecté, décrit ci-dessous).
   (f) Les sommets de $T$ admettent un degré au plus $k$ ?
9. Montrer que le problème Ensemble Domimant Connecté Connected dominating set est $\mathcal{NP}-$complet (Utiliser le problème Recouvrement de sommets).

   Ensemble Domimant Connecté (Ensemble Dominant Connecté)
   **Entrée :** $G = (V, E)$ un graphe non orienté, $k \in \mathbb{N}$.
   **Question :** Existe t'il un ensemble dominant $S$ dans $G$ de taille au plus $k$ tel que le sous-graphe de $G$ induit par les sommets de $S$ (*i.e.* $G_S = (S, E \cap S \times S)$ est connecté ?

10. Montrer que le problème 3-Satisfaisabilité Equilibré est $\mathcal{NP}-$complet.

   3-Satisfaisabilité Equilibré (3-Satisfaisabilité Equilibré)
   **Entrée :** Soit une formule logique dans laquelle chaque clause contient 3 littéraux et chaque variable apparaît un fois positivement et une fois négativement.
   **Question :** Existe-t'il une affectation qui satisfasse toutes les clauses ?

### 2.7.3 Autour des nombres

**Exercice 67 – Partition**

2-Partition (Partition)
**Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_n)$.
**Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

Monter que 2-Partition est $\mathcal{NP}-$complet.

**Exercice 68 – Autour du problème de la 2-Partition**

Nous rappelons que le problème suivant est $\mathcal{NP}$-complet.

2-Partition (Partition)
**Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_n)$ de somme $2P$.
**Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

Montrer que les problèmes suivants sont $\mathcal{NP}$-complets.

1. 2-PARTITION À VALEURS PAIRES (Partition à valeurs paires )

   **Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers à valeurs paires $p(a_1),\ p(a_2), \ldots,\ p(a_n)$ de somme $2P$.

   **Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

2. 2-PARTITION AVEC NOMBRE PAIRE (Partition avec nombre pair)

   **Entrée :** Etant donnés $2n$ objets $a_i$ $(1 \leq i \leq 2n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_{2n})$ de somme $2P$.

   **Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

3. 2-PARTITION ÉQUILIBRÉ (Partition équilibré)

   **Entrée :** Etant donnés $2n$ objets $a_i$ $(1 \leq i \leq 2n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_{2n})$ de somme $2P$.

   **Question :** Est-il possible de les partager en deux sous-ensembles $I$ et $\bar{I}$ de même poids total $P$ tel que $|I| = n$ ?

4. 2-PARTITION (Partition impair/paire)

   **Entrée :** Etant donnés $2n$ objets $a_i$ $(1 \leq i \leq 2n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_{2n})$ de somme $2P$.

   **Question :** Est-il possible de les partager en deux sous-ensembles $I$ et $\bar{I}$ tel que $\sum_{i \in I} a_i = \sum_{i \in \bar{I}} a_i$, avec un entier entre $a_{2j-1}$ et $a_{2j}$ appartenant à $I$ ?

5. 3-PARTITION (Partition à trois)

   **Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers à valeurs paires $p(a_1),\ p(a_2), \ldots,\ p(a_n)$ de somme $3P$.

   **Question :** Est-il possible de les partager en trois sous-ensembles $I_1,\ I_2$ et $I_3$ de $\{1, \ldots, n\}$ de même poids total $P$ ?

**Exercice 69 – Tri-partition**

3-PARTITION (3-Partition)

**Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_n)$.

**Question :** Est-il possible de les partager en trois sous-ensembles de même poids total $P$ ?

Montrer que 3-PARTITION est $\mathcal{NP}$-complet ?

**Exercice 70 – Programmation dynamique : algorithme pseudo-polynomial**

1. Sur le problème de la partition :

   2-PARTITION (Partition)

   **Entrée :** Etant donnés $n$ objets $a_i$ $(1 \leq i \leq n)$ de poids entiers $p(a_1),\ p(a_2), \ldots,\ p(a_n)$ de somme $2P$.

   **Question :** Est-il possible de les partager en deux sous-ensembles de même poids total $P$ ?

   (a) Nous allons plonger le problème dans une classe de problèmes dépendant de paramètres et liés par une relation de récurrence. On considère deux entiers $i$ et $j$ avec $1 \leq i \leq n$ et $0 \leq j \leq P$, et l'expression booléenne $T(i, j)$ : « étant donnés les $i$ premiers éléments de la famille, il existe un sous-ensemble de ces $i$ éléments de poids $j$ ». On remplit alors ligne par ligne un tableau $A$, qui contient les valeurs de $T$ dont les colonnes sont indicées par $j$ et les lignes par $i$.

   i. Donner la formule qui lie la ligne $i$ et $i-1$ et $p(a_i)$.
   ii. Illustrer ce principe avec les données suivantes : $n = 6,\ p(a_1) = 5, p(a_2) = 9,\ p(a_3) = 3,\ p(a_4) = 8,\ p(a_5) = 2,\ p(a_6) = 5$.
   iii. Comment avec le tableau rempli obtient-on les éléments de la partition ?

 (b) Donner la complexité de cet algorithme ?

2. Le problème du sac à dos :
   Nous considérons le problème du sac à dos sans répétition, c'est à dire les objets seront pris au plus une fois. Pour cela considérons, un tableau $K$ à deux dimensions tel que $K[j, w]$ représente la valeur maximale que l'on peut stocker dans un sac de capacité $w$ avec des objets $1, \ldots, j$.
   (a) Donner les formules ;
   (b) Illustrer le principe avec les données suivantes : $(w_1, v_1) = (1, 1); (w_2, v_2) = (2, 6); (w_3, v_3) = (5, 18); (w_4, v_4) = (6, 22); (w_5, v_5) = (7, 24)$ et $W = 12$.
   (c) Comment retrouver la solution à partir du tableau ?
   (d) Donner la complexité en temps et en mémoire

**Exercice 71 – Mètre du charpentier**

Montrer que le problème du mètre de charpentier est un problème $\mathcal{NP}$-complet

Mètre du charpentier (MC)
**Entrée :** La longueur de l'étui $L$ et des segments $l_i$ ($i$ de 1 à $n$).
**Question :** Peut-on plier le mètre pour qu'il rentre dans l'étui ?

**Exercice 72 – Algorithmes pseudo-polynomiaux**

Donner deux algorithmes pseudo-polynomiaux pour résoudre la problème du sac-à-dos dont le temps d'exécution est proportionnel au produit d'un polynôme en n (nombre d'objet) et au volume du sac à dos (pour l'un des algorithme) et au poids de l'objet le plus lourd (pour l'autre).

## 2.8 Classe $\mathcal{NL}$ & $\mathcal{NL}$-complet

**Exercice 73 – 2-Satisfaisabilité et $\mathcal{NL}$-complet**

Montrer que 2-Satisfaisabilité est $\mathcal{NL}$-complet. La preuve se fera à partir de Non-accessibilité.

Non-accessibilité (Unreachibility)
**Entrée :** $G = (V, E)$ et deux sommets $s, \quad t \in V$.
**Question :** Existe-t'il aucun chemin entre $s$ et $t$ ?

1. Rappeler le résultat reliant les classes $\mathcal{NL}$ et $co\mathcal{NL}$.
2. Montrer que 2-Satisfaisabilité appartient à $\mathcal{NL}$.
3. Procéder à la réduction entre Non-accessibilité et 2-Satisfaisabilité.

**Exercice 74 – Problème $\mathcal{NL}$=co$\mathcal{NL}$-complet**

$PATH = \{\langle G, s, t\rangle | G$ est un graphe orienté contenant un chemin du sommet s au sommet $t\}$ est $\mathcal{NL}$-complet. Pour montrer que $\mathcal{NL}$= co$\mathcal{NL}$, nous allons montrer que $PATH \in \mathcal{NL}$, en suivant les étapes ci-dessous. Voir $TD_1$1-corrigé

On peut montrer que 2-Satisfaisabilité appartient à $\mathcal{NL}$car peut être réduit (au sens de la réduction poly-log) à partir de GAP.

**Exercice 75 – Graphe fortement connexe et $\mathcal{NL}$-complétude**

Graphe fortement connexe
**Entrée :** $G = (V, E)$ un graphe orienté
**Question :** Est-ce que $G$ est fortement connecté ?

Montrer que Graphe fortement connexe est $\mathcal{NL}$-complet.

**Exercice 76 – Biparti et $\mathcal{NL}$**

Biparti
**Entrée :** $G = (V, E)$ un graphe
**Question :** Est-ce que $G$ est un graphe biparti ?

1. Caractériser un graphe avec la présence des cycles de longueurs impaires.
2. Montrer que Biparti est $\mathcal{NL}$.

## 2.9 Classe DP-complétude

**Exercice 77 – Réduction**

Considérons le problème suivant :

Best-Clique
**Entrée :** $G = (V, E)$ un graphe, $k \in \mathbb{N}$
**Question :** Est-ce que $G$ admet une clique de taille $k$ mais pas de clique de taille $k+1$ ?

On supposera que Best-Clique est NP-dur.

1. Montrer que Best-Clique est DP-complet.
2. Assumons que $\mathcal{NP} \neq$ co$\mathcal{NP}$. Montrer que Best-Clique $\notin \mathcal{NP}$.
3. Assumons que $\mathcal{NP} \neq$ co$\mathcal{NP}$. Montrer que Best-Clique $\notin$ co$\mathcal{NP}$.