Complexité & graphes : TD2

Université de Tours

`Département informatique de Blois`

NP : réduction et complétude

*
* *

Problème 1

Le problème Subset-Sum (somme de sous-ensemble) se définit comme suit :

- Subset-Sum :
  - *Instance* : Un couple (E, S) tel que :
    - Un ensemble $E = \{e_1, ..., e_n\}$ où $\forall i \in [\![1, n]\!], e_i \in \mathbb{N}$
    - Une valeur cible $S \in \mathbb{N}$.
  - *Question* : Existe t-il un sous-ensemble $E' \subseteq E$ tel que $S = \sum\limits_{e \in E'} e$?

On supposera (voir exercice n°4 pour le démontrer) que ce problème est NP-complet.

1. Le problème du sac à dos (Knapsack) se définit comme suit :
   - Knapsack :
     - *Instance* : Un triplet de la forme $(O, P_{\max}, V_{\min})$ tel que :
       - Un ensemble $O = \{o_1, ..., o_n\}$ d'objets où $\forall i \in [\![1, n]\!], o_i = (p_i, v_i) \in \mathbb{N}^2$. La valeur p_i désigne le poids de l'objet o_i et v_i sa valeur.
       - Une capacité maximale $P_{\max}$ du sac à dos (le poids maximal pouvant être supporté).
       - Une valeur minimale $V_{\min}$ du sac à dos (la valeur minimale des objets devant être transportés).
     - *Question* : Peut-on choisir un sous-ensemble $O' \subseteq O$ d'objets tel que : $\begin{cases} \sum\limits_{o \in O'} \pi_1(o) \leq P_{\max} \\ \sum\limits_{o \in O'} \pi_2(o) \geq V_{\min} \end{cases}$?

       Où π_k désigne la projection sur la k-ème coordonnée d'un vecteur.
       On a donc $\pi_1(o_i) = p_i$ et $\pi_2(o_i) = v_i$.

   Montrer que Subset-Sum est polynômialement réductible à Knapsack. En déduire que Knapsack est NP-complet.

   > On va réduire polynômialement Subset-Sum à Knapsack.
   >
   > Alors, Subset-Sum est polynômialement réductible à Knapsack si $\exists f$ calculable en temps polynomial et telle que pour tout couple (E, S), on a :
   >
   > $$(E, S) \in \text{Subset-Sum} \Leftrightarrow f(E, S) \in \text{Knapsack}$$

On pose f de la manière suivante : On crée une instance de Knapsack à partir des entrées de Subset-Sum telle que :

$$f(E,S) = (O^f, P^f_{\max}, V^f_{\min})$$

où :

- $O^f = \{(e_1,e_1),(e_2,e_2),...,(e_n,e_n)\}$ désigne l'ensemble d'objets issus de la fonction de transformation f.
- $P^f_{\max} = S$ désigne la capacité maximale du sac selon la fonction de transformation f.
- $V^f_{\max} = S$ désigne la valeur minimale du sac selon la fonction de transformation f.

Pour démontrer l'équivalence dans la formule de réduction, on raisonne par double implication :

- Implication 1 : $(E,S) \in$ Subset-Sum $\Rightarrow f(E,S) \in$ Knapsack

  On sait que $\exists E' \subseteq E$, un ensemble qui résout Subset-Sum. C'est-à-dire, on sait qu'il existe un ensemble E' d'éléments tel que $\sum\limits_{e \in E'} e = S$.

  On souhaite montrer que par notre fonction de tranformation f, alors si Subset-Sum est résolu, Knapsack l'est aussi. On souhaite donc montrer qu'il existe un sous-ensemble $O' \subseteq O^f$ d'objets tel que $\begin{cases} \sum\limits_{o \in O'} \pi_1(o) \leq P^f_{\max} \\ \sum\limits_{o \in O'} \pi_2(o) \geq V^f_{\min} \end{cases}$.

  On construit l'ensemble d'objets $O' = \bigcup\limits_{e \in E'} \{(e,e)\}$ dont tous les éléments e sont dans E'.

  Il est évident par contruction que $O' \subseteq O^f$.

  On fait la somme des poids et valeurs des objets de O' :

$$\begin{cases} \sum\limits_{o \in O'} \pi_1(o) \leq P^f_{\max} \\ \sum\limits_{o \in O'} \pi_2(o) \geq V^f_{\min} \end{cases}$$

  Or, on sait que $\forall o \in O', \pi_1(o) = \pi_2(o)$. Dès lors, on a :

$$\begin{cases} \sum\limits_{o \in O'} \pi_1(o) \leq P^f_{\max} \\ \sum\limits_{o \in O'} \pi_1(o) \geq V^f_{\min} \end{cases}$$

  De plus, on sait que $P^f_{\max} = V^f_{\min} = S$, on a donc :

$$\begin{cases} \sum\limits_{o \in O'} \pi_1(o) \leq S \\ \sum\limits_{o \in O'} \pi_1(o) \geq S \end{cases} \Leftrightarrow \sum_{o \in O'} \pi_1(o) = S$$

  Enfin, et par construction de O', on sait que $\sum\limits_{o \in O'} \pi_1(o) = \sum\limits_{e \in E'} e$. Il vient que :

$$\sum_{o \in O'} \pi_1(o) = \sum_{e \in E'} e = S$$

  L'égalité est bien vérifiée. Notre ensemble O' satisfait le problème Knapsack.

- Implication 2 : $(E,S) \in$ Subset-Sum $\Leftarrow f(E,S) \in$ Knapsack
  On sait que $\exists O' \subseteq O^f$ qui vérifie Knapsack, vérifions alors qu'il existe $E' \subseteq E$ qui vérifie Subset-Sum.
  On pose $E' = \{\pi_1(o) | o \in O'\}$. D'après la définition du problème Knapsack, il vient que

$$\begin{cases} \sum_{o \in O'} \pi_1(o) \leq P^f_{\max} \\ \sum_{o \in O'} \pi_2(o) \geq V^f_{\min} \end{cases}$$

  Or, comme $P^f_{\max} = V^f_{\min} = S$, on a donc :

$$\begin{cases} \sum_{o \in O'} \pi_1(o) \leq S \\ \sum_{o \in O'} \pi_2(o) \geq S \end{cases}$$

  De plus, comme $\forall o \in O^f, \pi_1(o) = \pi_2(o)$, il vient que $\sum_{o \in O'} \pi_1(o) = S$.
  Comme, $E' = \{\pi_1(o) | o \in O'\}$, alors $\sum_{e \in E'} e = \sum_{o \in O'} \pi_1(o) = S$
  Dès lors, l'implication est bien vérifiée. Notre ensemble E' satisfait le problème Susbset-Sum.

$\square$

Dès lors, on a Subset-Sum $\leq_p$ Knapsack.

Enfin, montrons que Knapsack est NP-complet.

Knapsack $\in$ NP car, étant donné une instance de problème $(O, P_{\max}, V_{\min})$ et un sous-ensemble $O' \subseteq O$, il est facile de vérifier les contraintes imposées.

Algorithme 1 : VerifKnapsack$(O', P_{\max}, V_{\min})$

```
Data : O' ⊆ O, P_max, V_min
Result : Boolean {true, false}
begin
    sumP ← 0 // Somme des poids
    sumV ← 0 // Somme des valeurs
    for o ∈ O' do
        sumP ← sumP + π1(o)
        sumV ← sumV + π2(o)
        // On vérifie que O' est un sous-ensemble de O
        if o ∉ O then
            return false
        O ← O − o
    if sumP ≤ P_max ∧ sumV ≥ V_min then
        return true
    else
        return false
```

La vérification se fait en un temps linéaire, en $O(n)$ donc Knapsack est bien NP.

Par réduction polynomiale, on a montré que Subset-Sum $\leq_p$ Knapsack. donc que Knapsack est NP-difficile (c'est à dire qu'il est au moins aussi difficile que tous les tous problèmes de NP, puisqu'il est au moins aussi difficile qu'un problème parmi les plus difficiles de NP).

Et puis enfin, puisque Knapsack est dans NP et NP-difficile, alors il est NP-complet (de manière à rappeler la définition de la NP-complétude).

2. Le problème de partition (PART) se définit comme suit :

   - PART :
     - *Instance* : Un ensemble $R = \{r_1, ..., r_n\}$ tel que $\forall i \in [\![1, n]\!], r_i \in \mathbb{N}$.
     - *Question* : Existe-t-il une partition R_1, R_2 de R telle que : $\sum_{r \in R_1} r = \sum_{r \in R_2} r$.
       On rappelle que R_1 et R_2 forment une partition de R si et seulement si $R_1, R_2 \subseteq R$ et que $R_1 \cap R_2 = \emptyset$ et $R_1 \cup R_2 = R$.

   Montrer que Subset-Sum est polynômialement réductible à PART. En déduire que PART est NP-complet.

On va réduire polynômialement Subset-Sum à PART.

Alors, Subset-Sum est polynômialement réductible à PART si $\exists f$ calculable en temps polynomial et telle que pour tout couple (E, S), on a $(E, S) \in$ Subset-Sum $\Leftrightarrow f(E, S) \in$ PART.

On pose f de la manière suivante : On crée une instance de Knapsack à partir des entrées de Subset-Sum telle que :

$$f(E, S) = R^f$$

où : $R^f = E \cup \{S + K, 2K - S\}$ avec $K = \sum_{e \in E} e$ la somme des éléments de E.

- Implication 1 : $(E, S) \in$ Subset-Sum $\Rightarrow f(E, S) \in$ PART

  On sait qu'il existe $E' \subseteq E$ tel que $\sum_{e \in E'} e = S$.

  Alors, R^f peut être partionné en deux sous-ensembles : $R_1 = E' \cup \{2K - S\}$ et $R_2 = (E \backslash E') \cup \{K + S\}$.

  R_1 et R_2 forment bien une partition de R^f, vérifions maintenant que :

$$\sum_{r \in R_1} r = \sum_{r \in R_2} r$$

  On a :
  - $\sum_{r \in R_1} r = \sum_{e \in E'} e + (2K - S) = S + (2K - S) = 2K$
  - $\sum_{r \in R_2} r = \sum_{e \in (E \backslash E')} e + (K + S) = \sum_{e \in E} e - \sum_{e \in E'} e + (K + S) = K - S + K + S = 2K$

  Dès lors, l'implication est bien vérifiée. Nos ensembles R_1 et R_2 satisfont le problème PART.

- Implication 2 : $(E, S) \in$ Subset-Sum $\Leftarrow f(E) \in$ PART

  On sait que R^f peut-être partitionné en deux sous-ensembles R_1, R_2 tels que $\sum_{r \in R_1} r = \sum_{r \in R_2} r$.

  On sait que $\sum_{r \in R^f} r = \sum_{e \in E} e + (2K - S) + (S + K) = K + 2K - S + S + K = 4K$

  Dès lors, comme $\sum_{r \in R_1} r = \sum_{r \in R_2} r$ et que R_1, R_2 forment une partition de R^f, on en déduit que :

$$\sum_{r \in R_1} r = \sum_{r \in R_2} r = 2K$$

Or $(S+K)+(2K-S)=3K$, donc ces deux éléments, $\{S+K\}$ et $\{2K-S\}$, ne peuvent pas être dans un même ensemble.

Admettons (sans perte de généralités) que $\{2K-S\} \in R_1$, alors :

$\sum_{r \in (R_1 \backslash \{2K-S\})} r = \sum_{r \in R_1} r - (2K-S) = 2K - 2K + S = S.$

Dès lors, on en déduit que $E' = R_1 \backslash \{2K-S\} \subseteq E$ qui satisfait bien Subset-Sum.

□

On a Subset-Sum $\leq_p$ PART. Dès lors, on sait que PART est NP-difficile, montrons également que PART est NP. Pour cela, on exhibe un algorithme qui vérifie une solution de PART en temps polynomial.

Algorithme 2 : VerifPART(R_1, R_2, R)

```
Data : R1, R2, R
Result : Boolean {true, false}
begin
    sumR1 ← 0 // Somme des éléments de R1
    sumR2 ← 0 // Somme des éléments de R2
    for r ∈ R1 do
        sumR1 ← sumR1 + r
        if ∃r' ∈ R2|r = r' then
            // On vérifie que r ∉ R2 : Vérification en O(|R2|)
            return false // R1 et R2 ne forment pas une partition
        R ← R − r
    for r ∈ R2 do
        sumR2 ← sumR2 + r
        if ∃r' ∈ R1|r = r' then
            // On vérifie que r ∉ R1 : Vérification en O(|R1|)
            return false
        R ← R − r
    if sumR1 = sumR2 ∧ R = ∅ then
        return true
    else
        return false
```

La complexité de cet algorithme est polynomiale en $O(|R_1| \times |R_2|)$ (où $|R_1|$ et $|R_2|$ sont la taille des partititions). Dès lors, PART $\in$ NP.

Ainsi, on a montré que PART est lui aussi NP-complet.

Problème 1 – Partie 2

On reprend ici les énoncés des problèmes précédents.

1. Sur Subset-Sum et Knapsack.

   (a) Soit la fonction de transformation $g(E,S) = (O^g, P^g_{\max}, V^g_{\min})$ telle que :

   - $O^g = \{(e_1,1),(e_2,1),...,(e_n,1)\}$
   - $P^g_{\max} = S$
   - $V^g_{\min} = n$

Montrer que g n'est pas valide pour montrer la réduction polynomiale de Subset-Sum à Knapsack.

(b) Soit l'instance de Subset-Sum suivante ($E = \{1, 2, 3, 4\}, S = 5$). On rappelle également la transformation $f(E, S) = (O^f, P^f_{\max}, V^f_{\min})$ utilisée dans la démonstration.

où :

- $O^f = \{(e_1, e_1), (e_2, e_2), ..., (e_n, e_n)\}$.
- $P^f_{\max} = S$.
- $V^f_{\max} = S$.

Montrer que l'instance $(E, S) \in$ Subset-Sum et que $f(E, S) \in$ Knapsack.

2. Sur Subset-Sum et PART.

(a) Soit la fonction de transformation $h(E, S) = R^h$ telle que $R^h = E \cup \{S\}$ qui transforme une instance de Subset-Sum en une instance de PART.

Montrer que h n'est pas valide pour montrer la réduction polynomiale de Subset-Sum à PART.

(b) Déterminer une fonction de transformation t et un exemple d'instance (E, S) tels que $(E, S) \notin$ Subset-Sum mais que $t(E, S) \in$ PART.

Problème 2

Le problème HC se définit comme suit :

- HC (Hamiltonian Circuit)
  - *Instance* : Un graphe non orienté $G = (N, A)$ où :
    - N est un ensemble de noeuds.
    - $A \subseteq N^2$ un ensemble d'arcs. On notera que, dans un graphe non orienté, $\forall (x, y) \in A$, on a aussi $(y, x) \in A$.
  - *Question* : Existe-t-il un circuit hamiltonien dans G ?

    On rappelle qu'un circuit hamiltonien $c = (n_1, ..., n_{|N|}, n_1)$ est un chemin qui part d'un noeud n_1, qui passe par tous les autres sommets de G une et une unique fois, puis revient en n_1.

  Ce problème est NP-complet (voir Exercice 3).

Le problème TSP ou *Problème du voyageur de commerce* se définit comme suit :

- TSP (Traveling Salesman Problem)
  - *Instance* : Un couple (G', D) où :
    - $G' = (S', (S')^2, \omega)$ est un graphe complet (dont tous les noeuds sont reliés ensemble) et où $\omega : S^2 \rightarrow \mathbb{R}^+$ est une fonction de coût sur les arcs (par exemple la distance entre deux villes).
    - $D \in \mathbb{R}^+$, un seuil de voyage.
  - *Question* : Existe-t-il un circuit hamiltonien $c = (n_1, ..., n_{|N|}, n_1)$ dans G' dont la distance de voyage n'excède pas D, c'est-à-dire tel que $\sum_{i=1}^{|N|} \omega(\pi_i(c), \pi_{i+1}(c)) \leq D$? (Problème de décision)[1]

[1] Parfois il s'agit de déterminer le circuit qui minimise le coût (Problème d'optimisation).

On va réduire polynômialement HC à TSP.
Alors, HC est polynômialement réductible à TSP si $\exists f$ calculable en temps polynomial et telle que pour tout G on a :

$$G \in \text{HC} \Leftrightarrow f(G) \in \text{TSP}$$

On pose f de la manière suivante : On crée une instance de TSP à partir des entrées de HC telle que :

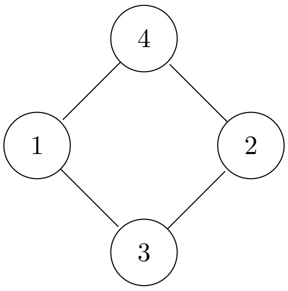
$$f(G) = (G^f, D^f)$$

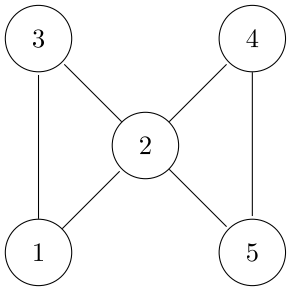
où :

- $G^f = (N, N^2, \omega^f)$ où N est l'ensemble des noeuds de G.
- $\omega^f(x,y) = \begin{cases} 0 & \text{si } (x,y) \in A \\ 1 & \text{si } (x,y) \notin A \end{cases}$.
- $D^f = 0$.

Pour démontrer l'équivalence dans la formule de réduction, on raisonne par double implication :

1. On considère le graphe G (à gauche) et G' (à droite) suivants en instance de HC :





   (a) Dessiner les graphes correspondant à $f(G)$ et $f(G')$.
   (b) Montrer que $G \in \text{HC}$ et que $f(G) \in \text{TSP}$.
   (c) Montrer que $G' \notin \text{TSP}$ et que $f(G') \notin \text{TSP}$.

2. Montrer que HC $\leq_p$ TSP.

3. En déduire que TSP est NP-complet.

Problème 3

Le problème DHC se définit comme suit :

- DHC (Directed Hamiltonian Circuit)
  - *Instance* : Un graphe orienté $G = (N, A)$ où :
    - N est un ensemble de noeuds.
    - $A \subseteq N^2$ un ensemble d'arcs.
  - *Question* : Existe-t-il un circuit hamiltonien dans G ?
    On rappelle qu'un circuit hamiltonien $c = (n_1, ..., n_{|N|}, n_1)$ est un chemin qui par d'un noeud n_1, qui passe par tous les autres sommets de G une et une unique fois, puis revient en n_1.

1. Montrer que HC $\leq_p$ DHC en notant que HC est un cas particulier de DHC.

2. On souhaite montrer à présent que DHC $\leq_p$ HC.

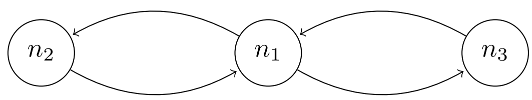
   Alors, DHC est polynômialement réductible à HC si $\exists f$ calculable en temps polynomial et telle que pour tout graphe dirigé G, on a :

$$G \in \text{DHC} \Leftrightarrow f(G) \in \text{HC}$$

   (a) Soit la fonction de tranformation $g(G) = G^g$ telle que $G^g = (N^g, A^g)$ avec :

   - $N^g = \bigcup_{n\in N} \left\{n^{(1)}, n^{(2)}\right\}$, c'est-à-dire que pour tout noeud $n \in N$ de G, on crée un ensemble de deux noeuds associés $n^{(1)}, n^{(2)}$.
   - $A^g = \bigcup_{n\in N} \left\{\left(n^{(1)}, n^{(2)}\right), \left(n^{(2)}, n^{(1)}\right)\right\} \cup \bigcup_{(n_i,n_j)\in A} \left\{\left(n_i^{(2)}, n_j^{(1)}\right), \left(n_j^{(1)}, n_i^{(2)}\right)\right\}$. on relie tous les noeuds $n^{(1)}, n^{(2)}$ de façon non dirigée et, pour les arcs qui existaient dans G entre n_i et n_j de la forme (n_i, n_j), on relie les noeuds $n_i^{(2)}$ et $n_j^{(1)}$ de façon non dirigée.

   On considère également le graphe G suivant en instance de DHC :



   i. Dessiner le graphe $g(G)$ résultant de la fonction de tranformation pour le graphe G ci-dessus.

   ii. Montrer que la transformation g qui transforme une instance de DHC en une instance de HC n'est pas valide pour montrer la réduction polynomiale de DHC à HC. En particulier, on montrera que $G \notin$ DHC mais que $g(G) \in$ HC.

   (b) Soit la fonction de tranformation $f(G) = G^f$ telle que $G^f = (N^f, A^f)$ avec :

   - $N^f = \bigcup_{n\in N} \left\{n^{(1)}, n^{(2)}, n^{(3)}\right\}$, c'est-à-dire que pour tout noeud $n \in N$ de G, on crée un ensemble de trois noeuds associés $n^{(1)}, n^{(2)}, n^{(3)}$.
   - $A^f = \bigcup_{n\in N} \left\{\left(n^{(1)}, n^{(2)}\right), \left(n^{(2)}, n^{(1)}\right), \left(n^{(2)}, n^{(3)}\right), \left(n^{(3)}, n^{(2)}\right)\right\} \cup \bigcup_{(n_i,n_j)\in A} \left\{\left(n_i^{(3)}, n_j^{(1)}\right), \left(n_j^{(1)}, n_i^{(3)}\right)\right\}$, on relie tous les noeuds $n^{(1)}, n^{(2)}, n^{(3)}$ de façon non dirigée et, pour les arcs qui existaient dans G entre n_i et n_j de la forme (n_i, n_j), on relie les noeuds $n_i^{(3)}$ et $n_j^{(1)}$ de façon non dirigée.

   Montrer que $G \notin$ DHC et que $f(G) \notin$ HC.

   (c) Grâce à la fonction f précédente, montrer que DHC $\leq_p$ HC.

   (d) En déduire que HC $\equiv_p$ DHC.

Problème 4

On rappelle que le problème 3-SAT est NP-complet en vertu du théorème de Cook-Levin.
L'objectif de cet exercice est de montrer que 3-SAT est polynômialement réductible à Subset-Sum, et d'en déduire que Subset-Sum est NP-complet.
Soit φ une instance de 3-SAT, c'est-à-dire une formule conjonctive telle que chaque clause de φ contient trois littéraux. On note M le nombre de clauses φ et N le nombre de variables. Alors :

$$\varphi = \bigwedge_{k=1}^{M} C_k$$

Où $C_k = l_{k,1} \vee l_{k,2} \vee l_{k,3}$ avec $l_{k,i} = x_i$ ou $\neg x_i$.
De plus, pour chaque variable x_i, on note $p(x_i) = \{k | x_i \in C_k\}$ c'est-à-dire l'ensemble des numéros des clauses où x_i apparait positivement et $n(x_i) = \{k | \neg x_i \in C_k\}$ c'est-à-dire l'ensemble des numéros des clauses où x_i est niée.

Soit la fonction f définie pour toute instance φ de 3-SAT par $f(\varphi) = (E, S)$ où :

- $E = \{p_1, n_1, ..., p_N, v_N, q_1, q'_1, ..., q_M, q'_M\}$ est un ensemble d'entiers de taille $2(M + N)$. Chaque élément $e_i \in E$ s'écrit sur $M + N$ chiffre en base 10.

  Les éléments de E sont définis comme suit :

  - $p_i = 10^{M+i} + \sum_{j \in p(x_i)} 10^j$
  - $n_i = 10^{M+i} + \sum_{j \in n(x_i)} 10^j$

  Les entiers p_i et n_i s'écrivent en base 10 avec $M + N$ chiffres égaux à 0 ou 1.

  De plus, à chaque clause C_k, on associe deux entiers q_k et q'_k tels que :

  - $q_k = q'_k = 10^k$

  Les entiers q_k et q'_k s'écrivent en base 10 avec M chiffres égaux à 0 ou à 1. Seul le chiffre à la position k est égal à 1 et tous les autres sont égaux à 0.

- $S = \sum_{i=1}^{N} 10^{M+i} + 3 \sum_{j=1}^{M} 10^j$

  L'entier S s'écrit en base 10 avec les chiffres 1 et 3. Son écriture en base 10 a la forme $\underbrace{11...1}_{N}\underbrace{33...3}_{M}$ où le premier bloc comporte N chiffres 1 et le second bloc comporte M chiffres 3.

1. Soit l'instance de 3-SAT telle que :

$$\varphi = C_1 \wedge C_2 \wedge C_3 = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_4)$$

   (a) Déterminer l'ensemble E et le nombre cible S par la fonction de transformation f proposée ci-dessus. On pourra compléter le tableau ci-dessous :

	x_4	x_3	x_2	x_1	C_1	C_2	C_3	Valeur
$p_1 =$	0	0	0	1	0	0	1	1001
$n_1 =$	0	0	0	1	1	0	0	1100
$p_2 =$								
$n_2 =$								
$p_3 =$								
$n_3 =$								
$p_4 =$								
$n_4 =$								
$q_1, q'_1 =$	0	0	0	0	0	0	1	1
$q_2, q'_2 =$								
$q_3, q'_3 =$								
$S =$								

(b) On considère la fonction d'interprétation $I(x_1, x_2, x_3, x_4) = (0, 1, 1, 1)$. C'est-à-dire que les variables x_2, x_3 et x_4 sont vraies et x_1 est fausse.

Montrer que l'on a bien $\varphi \in$ 3-SAT et que $f(\varphi) \in$ Subset-Sum.

2. Montrer que si $\varphi \in$ 3-Sat $\Rightarrow f(\varphi) \in$ Subset-Sum.
3. Montrer que $\varphi \in$ 3-Sat $\Leftarrow f(\varphi) \in$ Subset-Sum.
4. En déduire que 3-SAT $\leq_p$ Subset-Sum Subset-Sum et que Subset-Sum est NP-complet.

Partiel 2019 (5pts : 1.5 + 1 + 2.5)

On rappelle la définition du problème Partition (PART) :

- PART :
  - *Instance* : $R = \{r_1, ..., r_n\}$, un ensemble fini de nombres tel que $\forall i \in [\![1, n]\!], r_i \in \mathbb{N}$.
  - *Question* : Existe t-il une partition (R_1, R_2) de R telle que $\sum\limits_{r \in R_1} r = \sum\limits_{r \in R_2} r$?

Dans la suite, on suppose que PART est NP-complet.
On définit le problème Bin-Packing (BP) suivant :

- BP :
  - *Instance* : Un triplet (B, k, c) tel que $B = \{b_1, ..., b_n\}$ et deux entiers $k > 1$ et $c \geq 0$.
  - *Question* : Existe t-il une partition de B en k sous-ensembles $(B_1, ..., B_k)$ telle que $\forall i \in [\![1, k]\!], \sum\limits_{b \in B_i} b \geq c$.

On va réduire polynômialement PART à BP, c'est-à-dire, on cherche à montrer que :

$$\exists f \text{ polynomiale}, \forall R | R \in \text{PART} \Leftrightarrow f(R) \in \text{BP}$$

1. Montrer que la transformation $f(R) = (R, 2, 0)$ qui transforme une instance de PART en une instance de BP n'est pas valide pour montrer la réduction polynomiale de PART à BP.
2. On pose $R = \{1, 3, 8, 0, 6, 10, 4\}$. Montrer qu'ici $R \in$ PART et que $f(R) = \left(R, 2, \frac{1}{2}\sum\limits_{r \in R} r\right) \in$ BP .
3. En utilisant la fonction de transformation f précédente, démontrer que PART $\leq_p$ BP. En déduire que BP est NP-complet.