La NP-complétude et les entiers

Plan

Retour sur l'épisode précédent

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Faiblement/Fortement NP-complet

Stratégie pour prouver la NP-complétude

Pour prouver la NP-complétude d'un problème A, il suffit de prouver :

1. qu'il admet un vérificateur polynomial ;
2. que $B \leq A$ avec un problème connu NP-complet B

Somme de sous-ensembles

SOMME DE SOUS-ENSEMBLE
Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet.

Preuve :

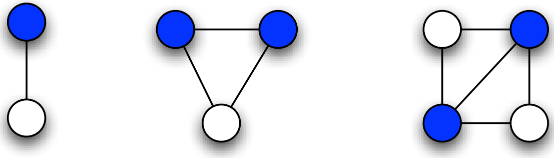
- SOMME DE SOUS-ENSEMBLE est dans NP.
- Couverture sommet $\leq$ SOMME DE SOUS-ENSEMBLE

Couverture sommet

Couverture de sommets (VC)
Données : un graphe non-orienté $G = (V, E)$ et un entier k.
Question : G contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus k ?

Théorème

Le problème Couverture de sommets est NP-complet.



Les sommets en bleu font partie de la couverture sommet

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou pas à la couverture et le sous-ensemble de sommets couvre bien toutes les arêtes.

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

- Pour cela,

  Exemple

  - on numérote les sommets entre 0 et $n - 1$
    et des arêtes entre 0 et $m - 1$.

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

- Pour cela,
  - on numérote les sommets entre 0 et $n-1$
    et des arêtes entre 0 et $m-1$.
  - on associe un entier b_{ij} pour chaque couple (arête,sommet)

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

- Pour cela,

  Exemple

  - on numérote les sommets entre 0 et $n-1$
    et des arêtes entre 0 et $m-1$.
  - on associe un entier b_{ij} pour chaque couple (arête,sommet)

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$

- On construit l'ensemble A de la façon suivante : $(b = 4)$

  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

- Pour cela,

  Exemple

  - on numérote les sommets entre 0 et $n-1$
    et des arêtes entre 0 et $m-1$.
  - on associe un entier b_{ij} pour chaque couple (arête,sommet)

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$

- On construit l'ensemble A de la façon suivante : $(b = 4)$
  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
  - Pour chaque arête j : est associé l'entier b^j

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A et définir t.

- Pour cela,

  Fin

  - on numérote les sommets entre 0 et $n-1$
    et des arêtes entre 0 et $m-1$.
  - on associe un entier b_{ij} pour chaque couple (arête,sommet)

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$

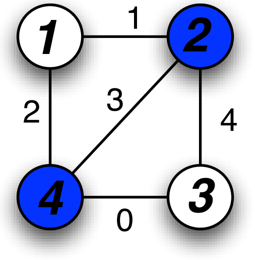
- On construit l'ensemble A de la façon suivante : $(b = 4)$
  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
  - Pour chaque arête j : est associé l'entier b^j
- On construit l'entier t :

$$t = \underbrace{k b^m}_{cardinalite\ \ de\ \ la\ \ couverture} + \sum_{i=0}^{m-1} \underbrace{2b^i}_{cout\ \ de\ \ l'\ \ arete\ \ i}$$

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



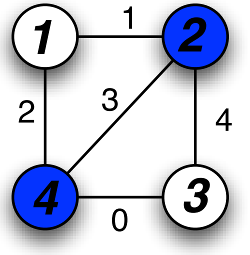
- $k = 2$

Back

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

◂ Back

L'ensemble A est composé des entiers suivants :

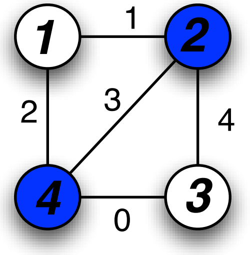
- a_1

(correspondant au sommet 1)

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est composé des entiers suivants :

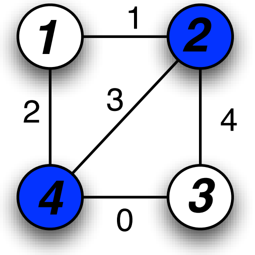
- $a_1 = b^5$ (correspondant au sommet 1)

Back

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est composé des entiers suivants :

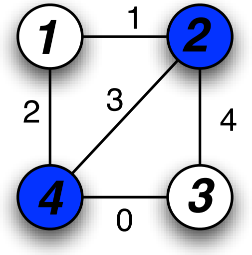
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)

Back

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est composé des entiers suivants :

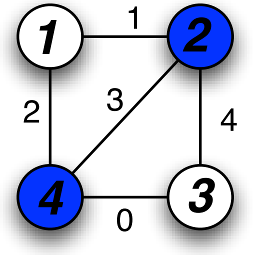
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ (sommet 2)
- $a_3 = b^5 + b^0 + b^4$ (sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ (sommet 4)

Back

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est composé des entiers suivants :

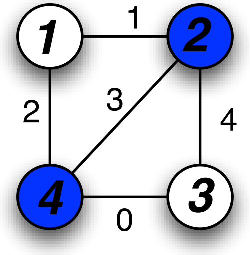
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ (sommet 2)
- $a_3 = b^5 + b^0 + b^4$ (sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ (sommet 4)
- b^0, b^1, b^2, b^3, b^4 (les entiers correspondants aux arêtes)

Back

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

Back

L'ensemble A est composé des entiers suivants :

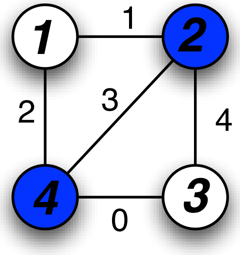
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ (sommet 2)
- $a_3 = b^5 + b^0 + b^4$ (sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ (sommet 4)
- b^0, b^1, b^2, b^3, b^4 (les entiers correspondants aux arêtes)

Si $S = \{2, 4\}$ est une couverture sommet, alors
$A' = \{a_2, a_4, \qquad \}$
Somme $= 2b^5 + \ b^0 + \ b^1 + \ b^2 + 2b^3 + \ b^4$

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

Back

L'ensemble A est composé des entiers suivants :

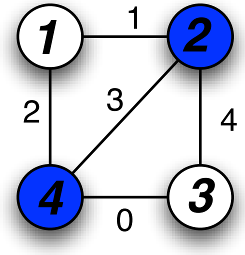
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ (sommet 2)
- $a_3 = b^5 + b^0 + b^4$ (sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ (sommet 4)
- $b^0,\ b^1,\ b^2,\ b^3,\ b^4$ (les entiers correspondants aux arêtes)

Si $S = \{2, 4\}$ est une couverture sommet, alors $A' = \{a_2, a_4, b^0, b^1, b^2, b^4\}$
Somme $= 2b^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

Back

L'ensemble A est composé des entiers suivants :

- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ (sommet 2)
- $a_3 = b^5 + b^0 + b^4$ (sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ (sommet 4)
- $b^0,\ b^1,\ b^2,\ b^3,\ b^4$ (les entiers correspondants aux arêtes)

$t = kb^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

Si $S = \{2, 4\}$ est une couverture sommet, alors
$A' = \{a_2, a_4, b^0, b^1, b^2, b^4\}$
Somme $= 2b^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

VC $\leq$ SOMME DE SOUS-ENSEMBLE (2/2)

- On peut prouver :

il existe une couverture sommet du graphe G de cardinalité k

$$\iff$$

il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$

VC $\leq$ SOMME DE SOUS-ENSEMBLE (2/2)

- On peut prouver :

  il existe une couverture sommet du graphe G de cardinalité k

  $$\Longleftrightarrow$$

  il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$

- La réduction se fait en temps polynomial :
  - Un entier est construit pour chaque arête et pour chaque sommet.

    Au pire des cas, il y a $\mathcal{O}(|E|)$ entiers dans l'instance.

  - Chaque entier se construit en $m + 1$ opérations.

Conclusion

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4.

Conclusion

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4 et en base 2.

Conclusion

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4 et en base 2.

Définitions :

- Un problème NP-complet est **faiblement NP-complet** si les entiers de l'instance sont codés en base 2.

Conclusion

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4 et en base 2.

Définitions :

- Un problème NP-complet est **faiblement NP-complet** si les entiers de l'instance sont codés en base 2.
- Un problème NP-complet est **fortement NP-complet** si les entiers de l'instance sont codés en unaire.

Le problème SOMME DE SOUS-ENSEMBLE,
est-il fortement NP-complet ?

Plan

Retour sur l'épisode précédent

Programmation dynamique
- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Faiblement/Fortement NP-complet

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Principe (Bellman, 1949)

Composer une solution optimale du problème en combinant les solutions (optimales) de ses sous-problèmes.

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Principe (Bellman, 1949)

Composer une solution optimale du problème en combinant les solutions (optimales) de ses sous-problèmes.

- En pratique :
  - Décomposer le problème en des sous-problèmes plus petits ;
  - Calculer les solutions optimales de tous ces sous-problèmes et les garder en mémoire.
  - Calculer la solution optimale à partir des solutions optimales des sous-problèmes

Plus précisément

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :
soit la récursivité
Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :
soit la récursivité
Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

nb exponentiel d'appels récursifs

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :

soit la récursivité

Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

nb exponentiel d'appels récursifs

ou soit la prog. dynamique :

Fib-Dynamique (n)

1. F[0]=1 ; F[0]=2
2. pour i allant de 3 à n, faire
   F[i]=F[i-1] + F[i-2] ;
3. retourner $F[n]$

n additions

Plus précisément

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

L'algorithme basé sur la programmation dynamique se compose de la façon suivante

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;
  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;
  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$
- Calculer les solutions optimales de tous ces sous-problèmes.
  Calculer la liste P_i des sommes des sous-ensembles de A_i.

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;
  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$
- Calculer les solutions optimales de tous ces sous-problèmes.
  Calculer la liste P_i des sommes des sous-ensembles de A_i.
- Calculer la solution optimale à partir des solutions optimales des sous-problèmes
  Trouver la relation entre les listes P_i et P_{i+1}

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \ldots, a_i\}$.

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les listes P_i :

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les listes P_i :
  - $P_0 =< 0 >$ (par convention)

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les listes P_i :
  - $P_0 =< 0 >$ (par convention)
  - $P_1 =< 0, 1 >$

listes des sommes distinctes des sous-ensembles ($\emptyset$, $\{1\}$) de $A_1 = \{a_1\}$

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les listes P_i :
  - $P_0 =< 0 >$ (par convention)
  - $P_1 =< 0, 1 >$

    listes des sommes distinctes des sous-ensembles $(\emptyset,\ \{1\})$ de $A_1 = \{a_1\}$
  - $P_2 =< 0, 1, 4, 5 >$

    listes des sommes distinctes des sous-ensembles $(\emptyset, \{1\}, \{4\}, \{1, 4\})$ de $A_2 = \{a_1, a_2\}$
  - $P_3 =< 0, 1, 4, 5, 6, 9, 10 >$

    listes des sommes distinctes des sous-ensembles de A_3
    $(\emptyset, \{1\}, \{4\}, \{1, 4\})$ union $(\emptyset \cup \{5\}, \{1, 5\}, \{4, 5\}, \{1, 4, 5\})$
    P_2 union $(\emptyset \cup \{5\}, \{1, 5\}, \{4, 5\}, \{1, 4, 5\})$

Notation

- P_i : la liste des sommes distinctes de tous les sous-ensembles de $A_i = \{a_1, \dots, a_i\}$.
- $L \oplus x$: la liste des entiers de L incrémentés par x

$$L \oplus x = \{s + x : s \in L\}$$

- On peut prouver : $P_i = P_{i-1} \cup (P_{i-1} \oplus a_i)$
- On note la procédure *fusion* $-$ *liste*(L, P) correspondant à la fusion de deux listes L et P d'entiers en une liste triée.

Cette procédure peut se réaliser en temps polynomial (en $\mathcal{O}(|L| + |P|)$ opérations)

Algorithme dynamique

Entrée : : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Sortie : un entier

1. $L_0 \leftarrow < 0 >$
2. pour i à allant de 1 à $|A|$
   2.1 $L_i \leftarrow \mathit{fusion} - \mathit{liste}(L_{i-1}, L_{i-1} \oplus a_i)$
   2.2 Supprimer tous les éléments de L_i qui sont supérieurs à t
3. retourner le plus grand élément de $L_{|A|}$

Algorithme dynamique

Entrée : : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Sortie : un entier

1. $L_0 \leftarrow < 0 >$
2. pour i à allant de 1 à $|A|$
   2.1 $L_i \leftarrow fusion - liste(L_{i-1}, L_{i-1} \oplus a_i)$
   2.2 Supprimer tous les éléments de L_i qui sont supérieurs à t
3. retourner le plus grand élément de $L_{|A|}$

Complexité : $\mathcal{O}(t \cdot |A|)$

- A chaque itération, la liste résultant contient au plus t éléments.

Algorithme dynamique

Entrée : : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Sortie : un entier

1. $L_0 \leftarrow < 0 >$
2. pour i à allant de 1 à $|A|$
   2.1 $L_i \leftarrow fusion - liste(L_{i-1}, L_{i-1} \oplus a_i)$
   2.2 Supprimer tous les éléments de L_i qui sont supérieurs à t
3. retourner le plus grand élément de $L_{|A|}$

Complexité : $\mathcal{O}(t \cdot |A|)$

- A chaque itération, la liste résultant contient au plus t éléments.

Remarque : On peut construire en même temps la solution.

Conclusion

L'algorithme retourne la plus grande valeur de la somme d'un sous-ensemble de A plus petit que t, en $\mathcal{O}(t \cdot |A|)$ opérations :

- en temps polynomial si l'entier en codé **en unaire**
- en temps exponentiel si l'entier en codé **en binaire**

  car $\mathcal{O}(t) = \mathcal{O}(2^{logt})$

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est faiblement NP-complet.

Plan

Problèmes fortement NP-complets

Il existe des problèmes fortement NP-complets comme :

SOMME DE SOUS-ENSEMBLE GÉNÉRALISÉ
Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$, en entier m et un entier $t \in \mathbb{N}$
Question : Existe-t-il m sous-ensembles A_1, ..., A_m deux à deux disjoints tels que $\displaystyle\sum_{a \in A_i} a = t$ pour $i = 1, \ldots, m$?

Autres problèmes fortement NP-complets.

Couverture de sommets pondérée (WVC)
Données : un graphe non-orienté $G = (V, E)$, une fonction de coût sur les sommets $c : V \rightarrow \mathbb{N}$ et un entier k.
Question : G contient-il une *couverture de sommets* $\mathcal{S}$ telle que $\sum_{v \in \mathcal{S}} c(v) \leq k$?

MAX-SAT
Données : un ensemble U de variables, un ensemble de clauses une fonction de coût sur les clauses et un entier k.
Question : Existe-t-il une fonction $t : U \rightarrow \{0, 1\}$ telle que la somme des coûts des clauses satisfaites par t est $\geq k$?