TD 11 : Approximation de Subset-Sum

1 Enoncé

On a un ensemble S = $\{x_1, \ldots, x_n\}$ d'entiers strictement positifs ainsi qu'un entier t . On cherche une somme partielle d'éléments de S maximale et inférieure à t. On appellera somme optimale ce nombre.

2 Premier Algorithme

Dans le premier algorithme , L_i représente l'ensemble des sommes partielles des éléments de $\{x_1, \ldots, x_i\}$ inférieures à t. L_n représente donc l'ensemble des sommes partielles de S inférieures à t. Par conséquent, max(L_n) renvoie exactement la somme optimale de S. De plus, cet algorithme a testé toutes les sommes partielles inférieures à t.

On en déduit donc que :

- La différence entre le résultat trouvé et la somme optimale est 0
- Dans le pire des cas, on teste toutes les sommes partielles de S soit une complexité de O(2^n)
- Si t = O($|S|^k$) pour k un entier fixé, alors $|L_i|$ = O($|S|^k$). Au pas i de la boucle, la fusion et la suppression se font en $O(|L_i|)$. Comme il y a n pas dans la boucle, la complexité totale est O($|S|^{k+1}$)

3 Deuxième Algorithme

3.1 Seuillage

Considérons une liste l = $\{y_0, \ldots, y_m\}$ d'entiers triés par ordre croissant et $\delta > 0$. L'opération de seuillage va consister à dire : soit y' le dernier élément gardé de l (on garde y_0 au départ) si $y_i > y' * (1 + \delta)$ alors on va garder y_i. En gros, on enlève des éléments qui sont trop proches d'autres proches d'autres éléments.

3.2 Algorithme

L'algorithme proposé va fonctionner de la même façon que le précédent sauf qu'il va seuiller à chaque pas de la boucle L_i par rapport à $\epsilon/2n$. On a alors $L_n = \{y_0, \ldots, y_m\}$. On va essayer de majorer m. On sait que :
$t \geq y_m \geq (1 + \epsilon/2n)^{m-1} * y_0 \geq (1 + \epsilon/2n)^{m-1}$

D'où : $m \leq \ln t / \ln(1 + \epsilon/2n) \leq (\ln t * 2n)/(\ln 2 * \epsilon) + 1$

m est donc polynomial en ln t et en $1/\epsilon$ $\ln t$ polynomial en la taille des données l'algorithme est en O(m * n) donc polynomial en la taille des données.

Il faut maintenant vérifier qu'on a bien trouvé une $(1+\epsilon)$ Approximation en montrant que :

$maxL_n \geq (1 + \epsilon) * Opt$(somme optimale)

On note P_i l'ensemble des sommes partielles de $\{x_1, \ldots, x_i\}$ inférieures à t.

Invariant : $\forall x \in P_i, \exists y \in L_i,\ x/(1 + \delta)^i \leq y \leq x$

Par récurrence :

- i = 0 : OK

- On suppose que c'est vrai pour (i-1) et on le montre pour i :

Soit $x \in P_i$

$P_i = P_{i-1} \cup (P_{i-1} + x_i)$

$x = x' + e$ avec $x' \in P_{i-1},\ e = 0\ ou\ e = x_i$

$x' \in P_{i-1}$ donc $\exists y',\ x'/(1 + \delta)^i \leq y' \leq x'$

Si y' + e est conservé par le seuillage :

$(x' + e)/(1 + \delta)^i \leq x'/(1 + \delta)^{i-1} + e \leq y' + e \leq x' + e = x$

Si y' + e n'est pas conservé par le seuillage :

$\exists y'' \in L_i,\ y'' \leq y' + x_i \leq y'' * (1 + \delta)$

$(y' + e)/(1 + \delta) \leq y'' \leq y' + e \leq x' + e \leq x$

$(x'/(1 + \delta)^{i-1} + e)/(1 + \delta) \leq y''$

$(x' + e)/(1 + \delta)^i \leq y''$

C'est l'encadrement recherché.

Grâce à l'invariant, on obtient : $Algo \geq Opt/(1 + \epsilon/2n)^n \geq Opt(1 + \epsilon)$

On a donc une $1 + \epsilon$ approximation de $Subset_sum$ polynomial en la taille des donnees et polynomial en $1/\epsilon$