Somme de sous-ensembles

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2013

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe.*** Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Introduction

Dans ce sujet, on va s'intéresser aux deux problèmes suivants :

Problème Somme de sous-ensemble : Soit un ensemble L de N entiers distincts et un entier t. Trouver une solution à l'équation

$$x_1 + \ldots + x_k = t \text{ pour } x_j \in L,$$

où les éléments de L apparaissent au plus une fois dans la somme.

Problème k-somme : Soit L_k, k ensembles de N entiers et un entier t. Trouver le nombre de solutions de l'équation

$$x_{1,i_1} + \ldots + x_{k,i_k} = t \text{ pour } x_{j,i_j} \in L_j.$$

Les parties 2, 3 et 4 sont relativement indépendantes.

1.1 Préliminaires

Dans la suite du sujet, on aura besoin de représenter des ensembles d'entiers. On notera

$$V(x, a, m, N) = \{v_k\}_{0 \le k < N} \text{ avec } v_0 = x \bmod m \text{ et } v_k = a \times v_{k-1} \bmod m, \text{ pour tout } k > 0.$$

Question 1 *Que valent :*
a) v_5 *dans* $V(u_0, 157, 1009, 6)$
b) v_{47} *dans* $V(u_0, 353, 997, 48)$
c) v_{86} *dans* $V(u_0, 1151, 16381, 87)$.

2 Somme de k-ensembles

2.1 Somme de 2-ensembles

Étant donné deux ensembles L_1 et L_2 de N entiers positifs avec multiplicité, c'est-à-dire qu'on n'enlèvera pas les doublons, on souhaite calculer le nombre de collisions, c'est-à-dire le nombre de couples (i, j) solutions de l'équation $x_{1,i} = x_{2,j}$ avec $x_{1,i} \in L_1$ et $x_{2,j} \in L_2$. On souhaite également calculer le nombre de solutions de l'équation $x_{1,i} + x_{2,j} = t$ avec $x_{1,i} \in L_1$ et $x_{2,j} \in L_2$ (on comptera également les multiplicités).

Question à développer pendant l'oral : Donner un algorithme pour résoudre ces deux problèmes. Déterminer sa complexité en temps et en mémoire.

Question 2 *Donner le nombre de collisions et le nombre de solutions de la somme de 2 ensembles pour les ensembles* L_1 *et* L_2 *:*
a) $L_1 = V(u_0, 157, 1009, 100), L_2 = V(u_0, 353, 997, 100), t = 112$
b) $L_1 = V(u_0, 157, 1009, 300), L_2 = V(u_0, 353, 997, 300), t = 233$
c) $L_1 = V(u_0, 157, 1009, 500), L_2 = V(u_0, 353, 997, 500), t = 185$.

2.2 Somme de 3-ensembles

Étant donné 3 ensembles L_1, L_2 et L_3 de N entiers positifs avec multiplicité, c'est-à-dire qu'on n'enlèvera pas les doublons, on souhaite calculer le nombre de solutions de l'équation $x_{1,i} + x_{2,j} + x_{3,k} = t$ avec $x_{1,i} \in L_1$, $x_{2,j} \in L_2$ et $x_{3,k} \in L_3$.

Question à développer pendant l'oral : Donner un algorithme pour résoudre ce problème et donner sa complexité en temps et en mémoire.

Question 3 *Donner le nombre de solutions pour le problème de la somme de 3 ensembles pour les ensembles L_1, L_2 et L_3 définis par :*
a) $L_1 = V(u_0, 157, 1009, 100), L_2 = V(u_0, 353, 997, 100), L_3 = V(13 \times u_0, 157, 1009, 100)$, $t = 233$
b) $L_1 = V(u_0, 157, 1009, 300), L_2 = V(u_0, 353, 997, 300), L_3 = V(24 \times u_0, 157, 1009, 300)$, $t = 133$
c) $L_1 = V(u_0, 1151, 16381, 4096), L_2 = V(67 \times u_0, 1151, 16381, 4096)$,
$L_3 = V(33 \times u_0, 1151, 16381, 4096), t = 48000$.

2.3 Somme de 4-ensembles

Étant donné 4 ensembles L_1, L_2, L_3 et L_4 de N entiers positifs avec multiplicité, c'est-à-dire qu'on n'enlèvera pas les doublons, on souhaite calculer le nombre de solutions de l'équation $x_{1,i} + x_{2,j} + x_{3,k} + x_{4,\ell} = t$ (on comptera également les multiplicités) avec $x_{1,i} \in L_1$, $x_{2,j} \in L_2$, $x_{3,k} \in L_3$ et $x_{4,\ell} \in L_4$.

Question à développer pendant l'oral : Donner un algorithme pour résoudre ce problème et donner sa complexité en temps et en mémoire

Question 4 *Compter le nombre de solutions de la somme de 4 ensembles pour les ensembles L_1, L_2, L_3 et L_4 définis par :*
a) $L_1 = V(u_0, 157, 1009, 100), L_2 = V(u_0, 353, 997, 100), L_3 = V(13 \times u_0, 157, 1009, 100)$,
$L_4 = V(13 \times u_0, 353, 997, 100), t = 3876$
b) $L_1 = V(u_0, 157, 1009, 300), L_2 = V(u_0, 353, 997, 300), L_3 = V(24 \times u_0, 157, 1009, 300)$,
$L_4 = V(24 \times u_0, 353, 997, 300), t = 3899$
c) $L_1 = V(u_0, 1151, 16381, 4096), L_2 = V(14 \times u_0, 1151, 16381, 4096)$,
$L_3 = V(33 \times u_0, 1151, 16381, 4096), L_4 = V(67 \times u_0, 1151, 16381, 4096), t = 65115$.

3 Somme de sous-ensemble

3.1 Problème exact

On s'intéresse dans cette partie au problème de somme de sous-ensemble de façon exacte. Soit un ensemble $L = \{x_0, \ldots, x_{N-1}\}$ de N entiers et un entier t cible. On cherche un sous-ensemble des valeurs de L dont la somme vaut t, c'est-à-dire

$$x_{i_1} + x_{i_2} + \ldots x_{i_k} = t \text{ pour } x_{i_j} \in L.$$

Considérons l'algorithme suivant SSE qui prend comme entrée L et t et renvoie la plus grande somme d'éléments d'un sous-ensemble de L inférieure ou égale à t :

1. $S_0 := \{0\}$
2. Pour $i = 1$ à N
   (a) $S_i := S_{i-1} \cup (S_{i-1} + x_{i-1})$ (avec multiplicité)
   (b) supprimer de S_i tout élément supérieur à t
3. **retourner** le plus grand élément de S_N

où les S_i sont des ensembles contenant les solutions possibles et $S_i + x$ représente l'ensemble $\{s + x : s \in S_i\}$.

Question 5 *Donner la somme la plus grande possible inférieure à t atteignable et le nombre de solutions (multiplicité dans S_N de cette valeur) pour*
a) $L = V(u_0, 353, 997, 10), t = 4000$
b) $L = V(u_0, 353, 997, 15), t = 4780$
c) $L = V(u_0, 353, 997, 16), t = 4900$.

Question à développer pendant l'oral : Quelle est la complexité en temps et mémoire de l'algorithme SSE ?

3.2 Problème approché

La version approchée est définie de la façon suivante : étant donné un ensemble de N entiers $x_0, x_2, \ldots, x_{N-1}$ et s, et on souhaite trouver un algorithme qui renvoie :
– oui, s'il existe un sous-ensemble dont la somme est s ;
– non, s'il n'y a pas de sous-ensemble dont la somme est entre $(1-\varepsilon)s$ et s pour un petit $0 < \varepsilon < 1$;
– oui ou non, s'il existe un sous-ensemble dont la somme est entre $(1-\varepsilon)s$ et s, mais aucun ne donne s.

On va considérer une opération de *seuillage* d'une liste L par rapport à δ. Elle consiste à supprimer le plus d'éléments possibles de L, de façon que si L' est le résultat du seuillage de L, alors pour tout élément y supprimé de L (c'est-à-dire $y \in L \setminus L'$), il existe encore un élément z de L' tel que $(1-\delta)y \leq z \leq y$. On peut voir z comme le représentant de y dans la nouvelle liste L'. Chaque y est représenté par un z tel que l'erreur relative de z par rapport à y est au plus égale à δ. Par exemple, si $\delta = 0.1$, et

$$L = \langle 10, 11, 12, 15, 20, 21, 22, 23, 24, 29 \rangle,$$

on peut obtenir

$$L' = \langle 10, 12, 15, 20, 23, 29 \rangle,$$

où 11 est représentée par 10, les valeurs 21 et 22 par 20 et la valeur 24 par 23.

Question à développer pendant l'oral : Supposons qu'on applique une opération de seuillage à la liste S_i pour $\delta = \varepsilon/N$. Décrire votre algorithme. On appellera S'_i l'ensemble résultant de cette suppression.

Question 6 *Donner la somme la plus grande possible inférieure à t pour*
a) $L = V(u_0, 998, 1000003, 100), t = 50000000, \varepsilon = 0.2$
b) $L = V(u_0, 998, 1000003, 100), t = 50000000, \varepsilon = 0.1$
c) $L = V(u_0, 998, 1000003, 200), t = 100000000, \varepsilon = 0.05$.

Question à développer pendant l'oral : Montrer par récurrence que pour tout élément de l'ensemble $y \leq t$ de S_i, il existe $z \in S'_i$ tel que

$$(1 - \varepsilon/N)^i y \leq z \leq y.$$

En déduire que la valeur retournée par l'algorithme n'est pas inférieure de plus de $1 - \varepsilon$ à la solution optimale y^*. Montrer alors que la taille des ensembles S'_i est bornée par une fonction de N, $\ln t$ et $1/\varepsilon$ et donner la complexité en temps et mémoire de votre algorithme.

3.3 Réduction de la mémoire pour la version exacte

Question à développer pendant l'oral : Donner un algorithme pour résoudre le problème de la somme de sous-ensembles de façon exacte en utilisant l'algorithme développé dans la partie 2.1 et donner sa complexité en temps et en mémoire.

Question 7 *Donner le nombre de solutions pour les problèmes :*
a) $L = V(u_0, 353, 997, 20), t = 4000$
b) $L = V(u_0, 353, 997, 24), t = 8000$
c) $L = V(u_0, 353, 997, 26), t = 10000$.

Question à développer pendant l'oral : Donner un algorithme pour résoudre le problème de la somme de sous-ensembles de façon exacte en utilisant l'algorithme développé dans la partie 2.3 et donner sa complexité en temps et en mémoire. Comment pourrait-on faire pour réduire la complexité en mémoire ?

Question 8 *Donner le nombre de solutions pour les problèmes :*
a) $L = V(u_0, 353, 997, 40), t = 16000$
b) $L = V(u_0, 353, 997, 44), t = 18000$
c) $L = V(u_0, 353, 997, 48), t = 20000$.

4 Stratégie optimale pour un jeu

Considérons un ensemble d'entiers positifs $v_0, \ldots, v_{N-1}$ pour N pair positionné dans cet ordre sur une ligne. On joue à un jeu contre un adversaire en jouant l'un après l'autre à chaque tour. À chaque tour, un joueur choisit soit le premier, soit le dernier nombre sur la ligne, l'enlève de la ligne de façon permanente et reçoit ce nombre. Déterminer la somme maximale que l'on est certain de gagner si on joue le premier.

Question à développer pendant l'oral : Donner une formule de récurrence permettant de calculer le gain garanti $V(i, j)$ si l'on considère les valeurs $v_i, v_{i+1}, \ldots, v_j$ si c'est à nous de jouer. En déduire un algorithme pour résoudre ce problème et donner sa complexité en temps et en mémoire.

Question 9 *Déterminer la somme si les valeurs v_i sont définies par :*
a) $V(u_0, 353, 997, 100)$ ***b)*** $V(u_0, 353, 997, 300)$ ***c)*** $V(u_0, 353, 997, 500)$.

Fiche réponse <u>type</u>: Somme de sous-ensembles

$\widetilde{\mathbf{u_0}}$: 26

Question 1

a) 571

b) 407

c) 5933

Question 2

a) (15,2)

b) (113,22)

c) (315,78)

Question 3

a) 43

b) 230

c) 9070

Question 4

a) 24

b) 851

c) 32817

Question 5

a) 3989, 1

b) 4780, 2

c) 4900, 8

Question 6

a) 49947692

b) 49980390

c) 99976236

Question 7

a) 253

b) 682

c) 50

Question 8

a) 12

b) 171

c) 1904

Question 9

a) 23968

b) 78581

c) $\boxed{123889}$

Fiche réponse: Somme de sous-ensembles

Nom, prénom, u_0: ....

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

Question 7

a)

b)

c)

Question 8

a)

b)

c)

Question 9

a)

b)

c)