

Option Informatique en Sup MPSI

Devoir surveillé du vendredi 15 avril 2005

Un problème d'optimisation combinatoire

Le problème de Samantha et Kevin

► Samantha et Kevin, qui viennent d'emménager dans un bel appartement¹, sont confrontés au délicat problème suivant : ils souhaitent installer une bibliothèque dans leur couloir, lequel mesure 437 cm de long ; le magasin propose des éléments de 37, 63 et 87 cm de long. L'idéal serait de trouver une combinaison d'éléments dont la somme des longueurs serait exactement 437 cm ; à défaut, une combinaison dont la somme des longueurs serait la plus grande possible, tout en étant inférieure à 437 cm. Pour son bagage mathématique, notre jeune couple n'a jamais eu besoin de porteur ; il se trouve donc fort désemparé face à ce délicat problème d'optimisation combinatoire. Vous allez les aider à le résoudre !

► Nous noterons t la longueur du couloir et $\ell = (\ell_1, \dots, \ell_k)$ la liste des longueurs des éléments proposés par le magasin. t et les ℓ_i sont des naturels non nuls. Nous dirons que $x \in \mathbb{N}$ est *SK-réalisable* avec ℓ s'il existe des naturels n_1, \dots, n_k tels que $x = \sum_{1 \leq i \leq k} n_i \ell_i$.

Question 1 • Montrez que x est SK-réalisable avec ℓ ssi l'une au moins des deux conditions suivantes est satisfaite :

- x est SK-réalisable avec $(\ell_1, \dots, \ell_{k-1})$
- $x - \ell_k$ est SK-réalisable avec (ℓ_1, \dots, ℓ_k)

Question 2 • Rédigez en Caml une fonction de signature :

```
SK_réalisable : int -> int list -> bool
```

spécifiée comme suit : `SK_réalisable x l` indique si x est SK-réalisable avec la liste ℓ . Objectif : trois lignes.

Question 3 ★★ • Montrez que le coût de l'exécution de la fonction précédente peut être exponentiel en la longueur $|\ell|$ de la liste ℓ ; l'unité de coût est l'opérateur `::`.

Question 4 • Dans cette question, nous supposons $\ell = (5, 7)$. Quel est le plus grand naturel qui n'est pas réalisable avec cette liste ?

Question 5 ★★ • Montrez que si deux des éléments de ℓ sont premiers entre eux, alors il existe $\xi \in \mathbb{N}$ tel que tout $x \geq \xi$ est réalisable avec ℓ .

► Un *graphe orienté* est un couple $G = (S, A)$ dans lequel G est un ensemble dont les éléments sont les *sommets* de G , et A est une partie de $S \times S$. Chaque élément de A est un *arc* : c'est donc un couple (u, v) d'éléments de S . Nous noterons $u \rightarrow v$ lorsque (u, v) est un arc de G . Un *chemin* de longueur n dans G est une suite $(t_i)_{0 \leq i \leq n}$ de sommets, telle que $t_{i-1} \rightarrow t_i$ pour tout $i \in \llbracket 1, n \rrbracket$.

► Nous pouvons résoudre le problème de Samantha et Kevin en construisant un graphe orienté dont les sommets sont des naturels ; (u, v) est un arc ssi il existe un indice $i \in \llbracket 1, k \rrbracket$ tel que $v = u + \ell_i$. La figure 1 montre une (petite) partie du graphe associé au problème de Samantha et Kevin ; chaque sommet est représenté par un cercle, et chaque arc par une flèche.

Question 6 • Décrivez un algorithme construisant un graphe orienté qui permettra de répondre au problème de Samantha et Kevin. Rappel : un algorithme doit rendre un résultat au bout d'un temps *fini*.

Question 7 • Finalement, Samantha et Kevin peuvent-ils meubler complètement leur couloir, avec les longueurs proposées par le magasin ?

¹d'environ 600 m²

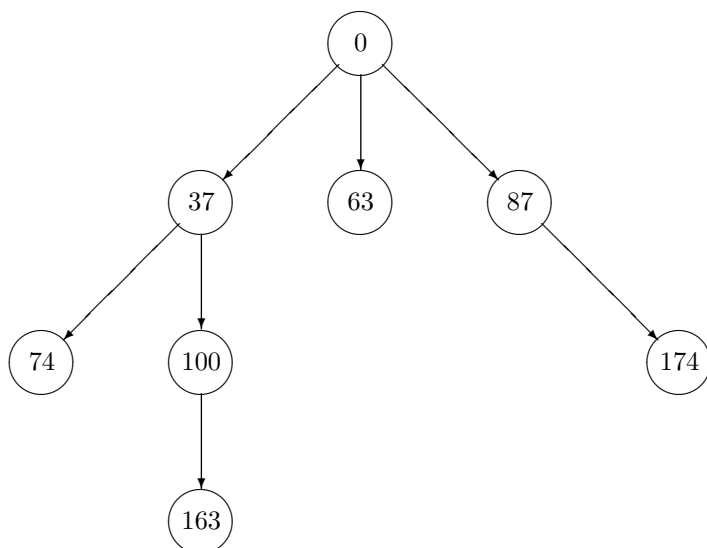


Figure 1: une partie du graphe orienté

Le problème de Gwendoline et Mikaël

► Samantha et Kevin ont donc réussi à installer leur bibliothèque ; lors de la pendaison de crémaillère, leurs amis Gwendoline et Mikaël leur expliquent qu'ils ont un problème analogue : ils viennent de changer d'appartement et veulent eux aussi monter une bibliothèque dans leur couloir, mais en réutilisant les éléments qu'ils avaient dans leur précédent logement. Samantha et Kevin leur suggèrent de consulter le spécialiste qui les a aidés si efficacement...

► Nous noterons $\ell = (\ell_1, \dots, \ell_k)$ les longueurs des éléments de la bibliothèque de Gwendoline et Mikaël. Nous dirons que x est *GM-réalisable* avec ℓ s'il existe une famille $(c_i)_{1 \leq i \leq k}$ de nombres égaux à 0 ou à 1, tels que $x = \sum_{1 \leq i \leq k} c_i \ell_i$. L'objectif est de déterminer le plus grand naturel au plus égal à t GM-réalisable avec ℓ .

Question 8 • Montrez que x est GM-réalisable avec ℓ ssi l'une au moins des deux conditions suivantes est satisfaite :

- x est GM-réalisable avec $(\ell_1, \dots, \ell_{k-1})$
- $x - \ell_k$ est GM-réalisable avec $(\ell_1, \dots, \ell_{k-1})$

Question 9 • Rédigez en Caml une fonction de signature :

```
GM_réalisable : int -> int list -> bool
```

spécifiée comme suit : `GM_réalisable x l` indique si x est réalisable avec la liste ℓ . Objectif : trois lignes.

Question 10 • Notons L la somme des longueurs des éléments dont disposent Gwendoline et Mikaël. Expliquez comment remplir une matrice M à $k + 1$ lignes (indexées de 0 à k) et $L + 1$ colonnes (indexées de 0 à L) avec les valeurs faux et vrai, de telle façon que $M_{i,j} = \text{vrai}$ ssi j est réalisable avec (ℓ_1, \dots, ℓ_i) .

Question 11 • Une fois la matrice M construite, comment Gwendoline et Mikaël obtiendront-ils la solution de leur problème?

► Subitement, Gwendoline se souvient d'un détail: ils avaient acheté leurs éléments au magasin *Leonardo*, lequel les vend par lots dont les longueurs sont des nombres de FIBONACCI consécutifs, les deux plus petits étant 1 et 2. De son côté, Mikaël note que le couloir du nouvel appartement est moins long que celui de l'ancien appartement.

Question 12 • Montrez qu'avec leurs éléments, ils peuvent remplir exactement le couloir du nouvel appartement.

Question 13 • Décrivez un algorithme *très simple* permettant à Gwendoline et Mikaël d'installer leur bibliothèque.

Question 14 • *PowerBib* est un autre magasin qui vend des étagères par éléments; les responsables du marketing souhaitent vendre, comme leur concurrent *Leonardo*, des paquets permettant de réaliser des bibliothèques de toute longueur au plus égale à 511 centimètres. Pour éviter d'avoir à verser des redevances à leur concurrent, ils font appel à vous pour concevoir une suite de valeurs réalisant cet objectif, et la plus courte possible. Que leur proposez-vous?

Annexe: recommandations pour l'écriture des programmes

► Chaque question de programmation fixe un *objectif*: c'est le nombre maximal de lignes que vous devez écrire. Vous compterez pour une ligne l'en-tête d'une fonction, comme:

```
let toto x y = fonction
```

Vous compterez chaque motif pour une ligne; notez bien qu'un motif union, comme:

```
| [] | [true] | [_;false;_] -> ...
```

compte pour une seule ligne.

► L'emploi de références ou de structures de données mutables est interdit.

FIN