

Exercice 1 (questions de cours / 04 points)

- 1) Dans un problème de minimisation d'une fonction f sur un espace de recherche S , que doit vérifier une solution optimale x^* ?
- 2) Pour appliquer la programmation dynamique à la résolution d'un problème d'optimisation, il faut diviser ce dernier en sous-problèmes imbriqués. Que doivent satisfaire ces sous-problèmes ?
- 3) Un algorithme glouton fournit-il toujours une solution optimale ? Justifier à l'aide d'un exemple vu au cours.
- 4) Rappeler les ingrédients de la méthode B&B (ce qu'il faut faire pour l'appliquer).

Exercice 2 (08 points)

L'algorithme ci-contre prend en entrée deux entiers k et M un tableau d'entiers C trié par ordre décroissant et retourne un tableau d'entiers R .

```
Int[] Algo(int k, int M, int[] C) {
    for (i = 0; i < k; i++)
        while (M >= C[i]) {
            M -= C[i];
            R[i]++;
        }
    return R; }
```

- 1) Calculer R pour $k=4$, $M=63$, $C = (25, 10, 5, 1)$
- 2) Rajouter la portion de code pour que cet algorithme retourne aussi la somme des éléments de R .
- 3) Evaluer la complexité de cet algorithme en notation O .
- 4) Cet algorithme permet de résoudre un problème d'optimisation combinatoire :
 - a) Qu'appelle-t-on ce problème ?
 - b) Quel est le type de cet algorithme ?
 - c) Cet algorithme fournit-il une borne supérieure ou une borne inférieure de l'optimum ? justifier.
 - d) Compléter cet algorithme afin qu'il retourne les deux bornes.

Exercice 3 (08 points)

Soit n et B deux entiers et un ensemble A de n entiers $a_i, i = 1, n$; et soit le problème d'optimisation P :

$$\begin{cases} \max_{x \in \{0,1\}^n} f(x) = \sum_{i=1}^n x_i a_i \\ \text{sujet à :} \\ \sum_{i=1}^n x_i a_i \leq B \\ x_i \in \{0, 1\} \end{cases}$$

- 1) Déterminer l'espace de recherche, sa taille, la variable de décision, la fonction objectif, les contraintes et le type de P .
- 2) Prenons $n = 4$, $B = 8$, $A = \{1, 4, 5, 6\}$
Soit la solution $x_0 = (1, 0, 0, 1)$. Montrer que x_0 est admissible puis calculer $f(x_0)$ (manuellement).
- 3) Prenons n , B et A quelconques.
 - a) Ecrire l'algorithme qui teste si une solution x est admissible.
 - b) Ecrire l'algorithme qui calcule $f(x)$ pour une solution x .
 - c) Ecrire l'équation récursive permettant de résoudre le problème P par la méthode de la programmation dynamique (l'algorithme n'est pas demandé).