

Programmation impérative et structures de données mutables

Exercice 1 : Implémentez la recherche dichotomique dans un tableau trié, dans un style impératif.

Exercice 2 : Écrire une fonction `tri` prenant en argument un tableau et une fonction de poids p et renvoyant le tableau trié suivant leurs poids par p .

Exercice 3 : On s'intéresse au nombre de surjections $u_{n,p}$ d'un ensemble à n éléments vers un ensemble à p éléments. La relation de récurrence qui relie les quantités $u_{n,p}$ s'écrit

$$u_{n,p} = p(u_{n-1,p-1} + u_{n-1,p})$$

1. Écrire une fonction qui calcule $u_{n,p}$ à l'aide d'une récursion standard. Quelle est sa complexité? La solution proposée est-elle optimale?
2. Écrire une fonction qui calcule $u_{n,p}$ de façon itérative en utilisant la programmation dynamique. Quelle est la complexité de cette dernière fonction?

Exercice 4 : Le problème du sac à dos.

On dispose de n objets, chacun ayant une masse $m_i \in \mathbb{N}^{+*}$ et une valeur $v_i \in \mathbb{R}^+$ pour $i \in \llbracket 1, n \rrbracket$. On cherche

$$P_{n,m} = \max \left\{ \sum_{i=1}^n x_i v_i, (x_1, \dots, x_n) \in \{0, 1\}^n \text{ tel que } \sum_{i=1}^n x_i m_i \leq m \right\}.$$

1. Montrer la relation $P_{k,m} = \max(P_{k-1,m}, v_k + P_{k-1,m-m_k})$ et en déduire une première fonction récursive `sac_a_dos_rec` qui prend en paramètre le tableau des masses, celui des valeurs et la masse à ne pas dépasser et qui renvoie la valeur maximale que peut contenir la sac à dos. Quelle est sa complexité?
2. Écrire une version gloutonne, dans laquelle on ajoute en premier l'objet dont le rapport v/m est maximal. Quelle est la complexité de l'algorithme? La solution obtenue est-elle optimale?
3. Proposer une version utilisant la programmation dynamique, qui renvoie la valeur maximale du sac à dos.
4. Écrire une fonction qui, à partir de la table des $P_{i,j}$ indique quels objets il faut prendre pour optimiser le sac à dos.