

DS 2

Option informatique, première année

Julien REICHERT

Toutes les fonctions de ce DS sont à écrire en Caml.

Exercice 1 : Écrire une fonction imprimant les instructions pour résoudre le problème des tours de Hanoi.

Remarque : Ce problème consiste à déplacer une pile de n (en argument) anneaux de taille croissante d'un tas (matérialisé par un piquet) à un autre (parmi trois), les opérations élémentaires étant le déplacement d'un anneau du haut d'une pile sur le haut d'une autre pile, à condition qu'il soit plus petit que l'ancien sommet de la pile d'arrivée. Le nombre optimal d'opérations élémentaires est $2^n - 1$.

Exercice 2 : Écrire un programme DPR qui calcule le nombre d'inversions dans un tableau (une inversion dans t est un couple (i, j) tel que $i < j$ et $t.(i) > t.(j)$) ou dans une liste (au choix, et de préférence).

Remarque : On justifiera le caractère DPR du programme en donnant la formule de calcul de la complexité, ainsi que la complexité proprement dite.

Exercice 3 : Réaliser une structure de file (écrire les opérations de base) à l'aide d'un tableau circulaire accompagné d'un triplet de références d'entiers mémorisant le nombre d'éléments dans la file et les positions de la tête et de la queue.

Exercice 4 : Réaliser une structure de dictionnaire (idem) à l'aide d'un tableau non trié dans lequel la suppression (on précisera alors la position de l'élément à supprimer) consiste à remplacer une entrée par le couple de chaînes vides et l'insertion récupère ces espaces libérés uniquement une fois que le reste du tableau est utilisé.

Exercice 5 : Résoudre le problème P1 ci-dessous (variante de la sous-somme) en se servant de solutions à des instances du problème P2 ci-dessous (variante du sac à dos).

Remarque : Le problème P1 est le suivant : on donne un ensemble d'entiers naturels tous distincts X ainsi qu'un entier naturel s et on demande s'il existe un sous-ensemble Y de X tel que la somme des éléments de Y soit s (sans chercher à déterminer un tel sous-ensemble). Le problème P2 est le suivant : on donne un ensemble de couples d'entiers naturels W , le premier élément de chaque couple étant la valeur et le deuxième étant le poids, ainsi qu'un entier naturel p et on demande la plus grande somme possible qu'on peut atteindre en additionnant des valeurs d'éléments différents de W , la somme des poids ne pouvant pas excéder p .

Indication : On considère l'existence d'une fonction `resout_p2 : (int * int) list -> int -> int` telle que `resout_p2 w p` retourne la valeur maximale v correspondante, et on demande d'écrire `resout_p1 : int list -> int -> bool` telle que `resout_p1 x s` détermine s'il existe une sous-liste de x de somme s .