Travaux Dirigés d'algorithmique n°3

Cours d'informatique de deuxième année

—DEUG Sciences S3-MI—

Représentation d'ensembles

L'objet du TD est la manipulation d'ensembles. Les ensembles seront représentés par des structures composées d'un tableau et du nombre d'éléments de l'ensemble. On procède dans un premier temps avec des tableaux non triés, puis avec des tableaux triés. On appréciera les différences de complexité entre les deux méthodes.

Nous allons manipuler des ensembles dont les éléments sont des personnes caractérisées par leur nom, leur prénom et leur âge.

1 Définition des types

On représente une chaîne de caractères par un tableau de `MAXSTR` caractères dont le dernier utile est '\0'. Donner un type de données pour représenter une personne. Donner un type de données pour représenter un ensemble de personnes d'au plus `MAXENS` éléments.

2 Ensembles

Dans cette section nous considérons des ensembles qui sont implémentés par une "structure non triée".

▸ **Exercice 1.** *Écrire une fonction permettant de saisir l'identité d'une personne au clavier. Écrire une autre fonction permettant l'affichage de l'identité d'une personne et une autre réalisant l'affichage d'un ensemble de personnes.*

▸ **Exercice 2.** *Écrire une fonction qui initialise un ensemble à l'ensemble vide.*

▸ **Exercice 3.** *Écrire une fonction qui prend en argument deux chaînes de caractères s_1 et s_2, qui retourne respectivement -1, 0 ou 1 suivant que $s_1 < s_2$, $s_1 = s_2$ ou $s_1 > s_2$, où $<$ est l'ordre lexicographique et $=$ l'égalité lexicographique.*

▸ **Exercice 4.** *Écrire une fonction qui prend en argument deux personnes p_1 et p_2, qui retourne respectivement -1,0 ou 1 suivant que $p_1 < p_2$, $p_1 = p_2$ ou $p_1 > p_2$, où $<$ est l'ordre naturel sur les personnes (obtenu en comparant d'abord les noms, puis les prénoms, puis les âges).*

▸ **Exercice 5.** *Écrire une fonction qui recherche une personne dans un ensemble de personnes, retourne l'indice de son apparition dans l'ensemble si elle y est présente,* `MAXENS` *sinon. Évaluer la complexité de cette fonction.*

▸ **Exercice 6.** *Écrire une fonction d'insertion d'une personne dans un ensemble de personnes. On rappelle qu'on ne peut avoir de doublons dans un ensemble. La fonction retournera* `VRAI` *si l'insertion s'est bien passée (pas de dépassement de la taille maximal d'un ensemble),* `FAUX` *sinon. Évaluer la complexité de cette fonction.*

▸ **Exercice 7.** *Écrire une fonction qui réalise la suppression d'une personne dans un ensemble de personnes. Évaluer la complexité de cette fonction.*

▸ **Exercice 8.** *Écrire deux fonctions qui réalisent respectivement l'union et l'intersection de deux ensembles de personnes. La fonction d'union retournera* `VRAI` *si l'union s'est bien passée (pas de dépassement de la taille maximal d'un ensemble),* `FAUX` *sinon. Évaluer la complexité de ces fonctions.*

3 Tri d'un ensemble

Définir un type de données pour représenter des couples d'indices d'éléments dans un tableau.

▸ **Exercice 9.** *Écrire une fonction* `minmax` *qui recherche à la fois la plus petite et la plus grande personne d'une partie $\{p_i, p_{i+1}, \ldots, p_{j-1}, p_j\}$ d'un ensemble de personnes $\{p_0, \ldots, p_{n-1}\}$, les indices i et j étant fournis à la fonction. Utiliser cette fonction* `minmax` *pour trier un ensemble de personnes. Calculer la complexité du tri obtenu.*

4 Encore des ensembles

Cette fois-ci nous allons représenter les ensembles par des "structures triées".

▸ **Exercice 10.** *Écrire une fonction de recherche "naïve" d'une personne dans un ensemble, et une autre qui pratique la recherche dichotomique. Comparer les deux fonctions.*

▸ **Exercice 11.** *Refaire les exercices 6, 7 et 8 de la section 2.*