Algorithmique et Structures de Données
TP 1 - Structure de données séquentielles et algorithmes de tri et de recherche

Halim Djerroud (hdd@ai.univ-paris8.fr)

Partie I

Touts les programmes suivants sont à écrire en C :

Les tableaux :

1. Écrire une fonction qui permet de demander à l'utilisateur de saisir une valeur entière et retourne cette valeur. La fonction doit redemander à l'utilisateur de ressaisir le nombre si la saisie n'est pas correcte.
2. Écrire une fonction qui prend en argument la taille du tableau, et créé dynamiquement le tableau, la fonction retourne un pointeur.
3. Écrire une fonction qui prend en argument un pointeur sur un tableau alloué dynamiquement et libère la mémoire allouée.
4. Écrire une fonction qui prend en paramètre un tableau et sa taille n, cette fonction doit remplir le tableau avec des entiers de *1* à n.
5. Écrire une fonction qui prend en argument un tableau et la taille du tableau, et affiche le contenu du tableau.
6. Écrire une fonction qui prend en argument un tableau et la taille du tableau, et remplie correctement le tableau avec des entiers saisie au clavier .
7. Écrire une fonction qui prend en argument deux tableau *t1* et *t2* de même taille, et la taille des tableaux, cette fonction doit copier les éléments de *t2* dans *t1*.
8. Écrire une fonction qui implémente la navette de knuth, cette méthode permet de mélanger un tableau d'entier d'une façon uniforme. l'algorithme est donné ci après :

```
Pour i allant de n  1 à 1 faire :
    j <- entier aléatoire entre 0 et i
    échanger a[j] et a[i]
```

Algorithmes de tri :

Pour chacune des questions suivantes, écrire des fonctions qui prend en argument un tableau et sa taille et renvoie le tableau trié.

1. Tri par sélection
2. Tri par insertion
3. Tri par bulles
4. Tri par fusion
5. Tri à peigne
6. Tri rapide
7. À l'aide des fonctions de l'exercice précédent générer un tableau aléatoire de 1000 entiers, appliquer l'ensemble des algorithmes sur le même jeu de données et mettre dans un tableau le temps d'exécution de chaque algorithme.
8. Faire la même chose en répétant l'expérience 100 fois, et donner les statistique pour chaque algorithme. Vous pouvez vous aider à l'aide du logiciel GnuPlot.

Algorithmes de recherche :

1. Écrire une fonction qui permet de rechercher un élément dans un tableau non trié
2. Écrire une fonction qui permet de rechercher un élément dans un tableau trié à l'aide de la méthode de recherche par dichotomie.

Allez plus loin :

1. Écrire une librairie (.so ou .a) qui permet de proposer l'ensemble des algorithmes implémentés (tri et recherche).
2. Écrire une documentation pour votre librairie.
3. Diffuser votre bibliothèque et la documentation dans votre *public_html*.

Partie II

Touts les programmes suivants sont à écrire en C++ :

Échauffement :

1. Écrire un programme qui affiche le message "Hello World !"
2. Écrire programme qui permet de lire deux entiers et calcule la somme.
3. Écrire une fonction qui permet permuter deux variables.

Les *templates* :

1. Écrire une fonction qui permet de calculer la sommes de deux nombres de type quelconque identiques (e.g. int, int).
2. Écrire une seconde fonction qui porte le même nom et qui permet de calculer la sommes de deux nombres de type quelconque différents (e.g. int,float). Inverser les les variables lors de l'appel de la fonction ; Que constatez vous ?

Les *vectors* :

1. Écrire un programme qui déclare un vecteur (vec) d'entiers et insérer 10 valeurs et finalement afficher le contenu de contenu à l'aide d'une boucle `for`.
2. Déclarer un vecteur de 10 valeurs initialisé à 5 ;

Les algorithmes :

1. Écrire un programme qui déclare un vecteur (vec) d'entiers de 10 valeurs et remplir avec des valeurs de 1 à 10 avec une boucle `for`.
2. Utiliser l'algorithme (fonction) `random_shuffle` pour mélanger le vecteur.
3. trouver le maximum et le minimum.
4. Trier le vecteur.