

**TD2**

**Prof. Habiba Drias**

## Exercices

### Exercice 2.1

- 1) Implémenter chacune des structures suivantes à l'aide de tableaux
  - a. Liste
  - b. Queue
  - c. Pile
- 2) Ecrire des algorithmes d'insertion et de suppression d'un élément en tenant compte de tous les cas possibles de l'état de la structure à savoir : vide, contenant un seul élément ou contenant plus d'un élément. Calculer la complexité de chacun des algorithmes développés.
- 3) Répéter les deux premières questions en utilisant des listes dynamiques à la place des tableaux

### Exercice 2.2

Prendre l'exercice précédent mais en considérant cette fois-ci, la structure d'arbre binaire. Calculer la complexité de chacun des algorithmes développés.

### Exercice 2.3

Considérer un tableau constitué des  $n$  premiers nombres entiers  $(1, 2, \dots, n)$ . Une méthode de détermination des nombres premiers inférieurs au sens large à  $n$ , consiste à considérer le nombre 2 qui est premier puis à éliminer tous les nombres multiples de 2 car ils ne sont pas premiers, ensuite itérer ce processus en continuant avec le nombre suivant non éliminé c'est-à-dire 3 jusqu'à traiter tous les entiers du tableau.

- 1) Illustrer chaque itération du procédé de calcul des nombres premiers décrit ci-dessus sur les 10 premiers nombres entiers
- 2) Ecrire un algorithme de calcul et d'impression des nombres premiers inférieurs au sens large à  $n$ . Il est recommandé par souci de simplification de l'algorithme de mettre le nombre à 0 s'il est premier et à 1 sinon, une fois qu'il est traité.
- 3) Calculer la complexité de l'algorithme.
- 4) Ecrire l'algorithme en assembleur à l'aide du jeu d'instructions donné en cours.
- 5) Calculer la complexité du programme assembleur

### Exercice 2.4

Ecrire les procédures suivantes :

- Insertion d'un élément dans un ensemble
- Suppression d'un élément d'un ensemble
- Union de deux ensembles
- Intersection de deux ensembles

### Exercice 2.5

Le parcours des nœuds dans un arbre binaire peut se faire selon les deux stratégies suivantes :

- Le parcours selon l'ordre préfixé défini comme suit :
  - a. Visiter la racine
  - b. Parcourir selon l'ordre préfixé le sous arbre dont la racine est le fils gauche de la racine

- c. Parcourir selon l'ordre préfixé le sous arbre dont la racine est le fils droit de la racine
- Le parcours selon l'ordre post-fixé défini comme suit :
- a. Parcourir selon l'ordre post-fixé le sous arbre dont la racine est le fils gauche de la racine
  - b. Parcourir selon l'ordre post-fixé le sous arbre dont la racine est le fils droit de la racine
  - c. Visiter la racine
- 1- Pour chacune de ces stratégies, écrire un algorithme récursif pour parcourir un arbre binaire.
  - 2- Donner les versions itératives de ces algorithmes

### **Exercice 2.6**

- 1- Ecrire un algorithme pour inverser l'ordre d'une liste. Calculer sa complexité
- 2- Ecrire un algorithme pour créer une liste doublement chaînée à partir d'une liste quelconque. Calculer sa complexité

### **Exercice 2.7**

Considérer deux listes de nombres entiers triées par ordre croissant. On souhaiterait fusionner ces deux listes pour obtenir une troisième liste triée.

- 1) Ecrire un algorithme de fusion de deux listes triées
  - a. En utilisant la structure de tableau
  - b. En utilisant la structure dynamique de listes
- 2) Quelle est la complexité de votre algorithme.
- 3) Comment représenter une liste chaînée en assembleur. Illustrer votre réponse à l'aide d'un petit exemple
- 4) Ecrire l'algorithme de fusion proposé en assembleur à l'aide du jeu d'instructions donné en cours.
- 5) Calculer sa complexité

### **Exercice 2.8** (Tour de Hanoi)

Ecrire une version itérative de la procédure de la tour de Hanoi vue en cours.

### **Exercice 2.10**

Ecrire un algorithme pour évaluer les expressions arithmétiques sur les opérateurs + et \*, écrites en

- a) notation polonaise (préfixée)
- b) infixée
- c) post-fixée

### **Exercice 2.11**

Soit  $G = (V, E)$  un graphe orienté sans cycle. Ecrire un algorithme pour associer des entiers aux sommets de  $G$  de manière que si  $(v, w)$  est un arc orienté alors l'entier associé à  $v$  est plus petit que l'entier associé à  $w$ . Calculer sa complexité ? (sol:  $(|E| + |V|)$ ).

### **Exercice 2.12**

Soient  $G = (V, E)$  un graphe orienté sans cycle et deux sommets  $s$  et  $d$  de  $V$ . Ecrire un algorithme pour trouver un ensemble de chaînes de  $s$  vers  $d$  satisfaisant les conditions suivantes:

- 1) Seuls  $s$  et  $d$  sont communs à deux chaînes.
  - 2) Aucune autre chaîne satisfaisant (1) ne peut être ajoutée à l'ensemble.
- Quelle est la complexité de l'algorithme ? (Sol:  $(|E| + |V|)$  .