

STRUCTURES DE DONNÉES ABSTRAITES

Exercice 1

Une file avec deux piles

Il est possible d'implémenter une file à l'aide de deux piles (A et B) de la manière suivante : l'ajout des éléments se fait dans la pile A et leur retrait depuis la pile B . Lorsque l'on souhaite retirer un élément alors que la pile B est vide on déplace d'abord les éléments de la pile A vers la pile B .

1. Proposer une implémentation des opérations de manipulation d'une file selon ce principe.
2. Donner la complexité temporelle des différentes opérations.
3. On considère l'ajout et le retrait de n éléments de la file. Quelle est alors la complexité temporelle par élément des retraits ?

Exercice 2

Parenthèses

Une séquence de parenthèses (au sens large : on accepte tous les signes de parenthésage) est correcte si et seulement si 1) chaque parenthèse ouverte est fermée ensuite et 2) une parenthèse n'est fermée que lorsque toutes les parenthèses ouvertes après elle ont été fermées. Par exemple la séquence $(\{(\{())\})$ est correcte mais pas la séquence $\{()\{()\}$ ni la séquence $(\{)\}$. On souhaite vérifier si une séquence de parenthèses est correcte.

1. Proposer des structures de données adaptées à la résolution de ce problème.
2. Écrire un algorithme vérifiant qu'une séquence de parenthèses est correcte.

Exercice 3

Tri selon un ordre partiel

On se donne un ensemble (E) d'éléments ainsi qu'une relation d'ordre partielle ($R \subseteq E \times E$) sur ces éléments. On souhaite trier les éléments de E , c'est à dire donner une séquence $e_1 \dots e_n$ contenant exactement une fois chaque élément de E et telle que $\forall i < j, e_i R e_j$ ou e_i et e_j ne sont pas en relation.

1. Proposer une structure de données adaptée à la résolution de ce problème.
2. Écrire un algorithme triant un ensemble selon un ordre partiel.

Exercice 4

Aéroport

On considère un aéroport. Plusieurs avions souhaitent atterrir. Il existe différentes contraintes (quantité de carburant restant, correspondances des passagers...) qui définissent l'ordre dans lequel l'autorisation leur sera donnée. On souhaite écrire un programme qui décide de l'ordre dans lequel les avions doivent atterrir au fur et à mesure que ceux-ci arrivent aux abords de l'aéroport.

1. Proposer une structure de données adaptée à la résolution de ce problème.
2. Écrire un algorithme gérant l'ordre d'atterrissage des avions.

Exercice 5

Quelques propriétés des arbres binaires

1. Montrer que le nombre maximal de nœuds à profondeur k dans un arbre binaire est 2^k .
2. Soit un arbre binaire à n nœuds, montrer que le nombre de feuilles f de cet arbre est tel que $f \leq n - f + 1$.

Soit la fonction suivante, qui parcourt un arbre binaire :

```
1: parcours A =
2:     si A est vide alors terminer
3:     sinon
4:         parcours G(A)
5:         parcours D(A)
```

Si on ajoute un affichage de la racine de A à différentes positions dans cette fonction, on obtient la liste des sommets de l'arbre sur lequel on appelle la fonction :

- en ordre préfixe si l'affichage est fait entre la ligne 3 et la ligne 4,
- en ordre infixé s'il est fait entre les lignes 4 et 5,
- ou en ordre suffixé s'il est fait après la ligne 5.

3. Montrer que si on connaît la liste des nœuds d'un arbre binaire en ordre préfixé et en ordre infixé on peut reconstruire l'arbre. Est-ce toujours vrai si on connaît la liste des nœuds en ordre préfixé et en ordre suffixé ? En ordre infixé et en ordre suffixé ?

Exercice 6

Deux piles dans un tableau

1. Comment peut-on représenter deux piles par un seul tableau en assurant que, tant que le tableau n'est pas plein, il est possible d'ajouter un élément dans n'importe quelle pile ?
2. Proposer une implémentation des opérations de base des piles dans ce cas.