Interface PM
IFT2905
Fuille de notes

Franz Girardin

20 février 2024

Table des matières

Section 1

Introduction aux structures de données

1.1 Caractéristique génériques de 𝕊𝔻

Comment décrire une structure de données

- ▷ **Linéaire ou *non* linéaire**
  - ► P. ex. `graph` vs. `arrays`
- ▷ Homogène ou non homogène
  - ► Indique si les données sont mê `type`
- ▷ Statique ou *dynamique*
  - ► P. ex. taille modifiable ou `fixe`.

Note :

Plusieurs problèmes ont une complexité telle que les 𝕊𝔻 **base** sont inappropriées, d'où la nécessité des 𝕊𝔻 **abstraites**, plus adéquates.

𝕊𝔻 Abstaites Elles `généralise` les types de données de base et permettent l'émergence de concept sophistiqués d'**abstraction**, d'**encapsulation** et de typage. Elle possèdent **5 caractéristiques** fondamentales (𝕋𝕌𝕆ℙ𝔸) :

- ▷ Type
- ▷ Utilise
- ▷ 𝕆pérations
- ▷ ℙrocédures
- ▷ Axiomes

1.2 Tableau

Stocke `E` dans emplacement de mémoire contigus.

- ▷ De longueur fixe ou variable
  - ► Autrefois : adresse brutes
  - ► Maintenant : N `arrays` (indexation)
  - ► Maintenant : `records` (champs)

1.3 Pile

La propriété fondamentale d'une pile est que le seul éléement accessible et celui se trouvant au sommet de la pile. La **pile** suit le principe **LIFO** : *last in, first out.*

1.4 File

La **file** suit le principe **FIFO** : *first in, first out.* Elle possède, entre autres, les opération enfiler *enqueue* et défiler *dequeue.*

1.5 Liste

Une file est le résultat de l'*ordonancement* d'un nombre dénombrable de données où la même valeur peut apparaître plusieurs fois.

- ▷ **Implémentation**
  - ► Liste chaînée `Linked List`
  - ► Liste doubleemnt chaînée `Linked List`
  - ► Liste circulaire
  - ► Tableau `array`

Note :

Une liste chaînée stocke un ensemble d'éléments de façon linéaire. Chaque élément ou nœud d'une liste chaînée contient un **élément de données** ainsi qu'une **référence**, ou lien, vers l'élément suivant de la liste.

1.6 Arbre

Un arbre stocke un ensemble d'éléments sous une forme **hiérarchique abstraite**. Chaque nœud est relié aux autres et peut contenir plusieurs sous-valeurs appelées enfants. Il s'agit d'un `graphe` avec **3 particularité fondamentales** :

- ▷ Acyclique
- ▷ Connexe
- ▷ Possède une seule racine

1.7 File de priorité

Possède les mêmes propriété de base qu'un file, sauf que chaque élément a en plus un poids de priorité qui détermine l'ordre global des éléments.

1.8 Graphe

Un graphe stocke un ensemble d'éléments de façon non linéaire. Il se compose d'un ensemble fini de nœuds, appelés **sommets**, et de lignes, les arêtes, qui relient les sommets entre eux. Les graphes permettent notamment de représenter des systèmes réels, comme des réseaux informatiques.

Théorème des quatre couleurs Il existe une 4-coloration de n'importe qu'elle pays qui a été découpé sur une carte.

Section 2

Pile

Exemples d'applications

- ▷ Historique de pages visitées
- ▷ Annuler une séquence de texte dans un éditeur
- ▷ Chaîne de méthodes appelées d'un fonction complexe

5 Opération fondamentales

- ▷ Créer un pile
- ▷ Empiler
- ▷ Dépiler
- ▷ Regarder le sommet
- ▷ Obtenir le nombre d'éléments

```
elements[0...n-1] <-- tableau // tableau taille n
// top <-- 0 pour assigner 0 Ã  top on fait :

push(E) :
element[top] <-- E
top <-- top + 1 // modifie l'index du sommet


//  Pour obtenir l'E au sommet, on fait :

pop() :
retourne element[top]
top <-- top -1 //decremente la valeur du sommet
x <-- element[top + 1] // enregistre val sommet
element[top + 1] <-- null
```

Conventions d'écriture

- ▷ **IN**fixé : `gauche racine droite`
  - ► $5 - 6\times 7 \leftrightarrow (5-6)\times 7$
- ▷ **POST**fixé : `gauche droite racine`
  - ► $56-7\times \leftrightarrow (56-)\ 7\times$
- ▷ **PRÉ**fixé : `racine gauche droite`
  - ► $\times -567\times \leftrightarrow \times(-56)\ 7$