

Introduction

Structures de  
données  
linéaires

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Structures de données et algorithmes

Joëlle Cohen

7 avril 2016

Introduction

Structures de  
données  
linéaires

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Plan I

## ① Introduction

## ② Structures de données linéaires

- Généralités
- Les tableaux
- Autres structures linéaires : un exemple
- Les piles
- Exercices
- les files
- Exercices
- Les listes
- Exercices

## ③ Structures de données arborescentes

- les arbres binaires
- Exercices
- Parcours des arbres binaires
  - Parcours en profondeur
  - Parcours en largeur
- Arbres binaires de recherche

Introduction

Structures de  
données  
linéaires

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

- Arbres équilibrés AVL
  - Les files de priorité
- Exercices

## Plan II

- 4 Dictionnaire et table de hachage
  - Dictionnaire
  - Tables de hachage

## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

Algorithme : méthode de résolution d'un problème

Structure de données : méthodes pour stocker et manipuler des données.

Algorithme + Structure de données = programme.

Pour un problème donné, il peut y avoir plusieurs algorithmes qui peuvent utiliser des structures de données différentes d'un algorithme à l'autre.

Par exemple un tri peut utiliser un tableau (tri à bulle, quicksort, ...), une liste chaînée (tri fusion), un arbre binaire de recherche, un tas (tri par tas).

## Introduction

Structures de données  
linéaires

Structures de données  
arborescentes

Dictionnaire et  
table de hachage

Documenter  
un projet

Le but du cours est de savoir manipuler un ensemble de données selon un contexte imposé (par le problème ou par l'algorithme résolvant ce problème).

Pour un problème donné, on cherche à déterminer la meilleure façon de manipuler les données mises en jeu dans ce problème.

Pour cela on va

- *spécifier* = construire un modèle abstrait du réel et décrire ce modèle en vue d'assurer l'adéquation entre ce qui est produit et ce qu'on attend,

- *utiliser* une structure de données

*structure de données* = modèle qui décrit le comportement d'un ensemble d'informations muni de ses propriétés.

Une structure de données peut être vue de l'extérieur (modèle abstrait) et de l'intérieur (réalisation concrète).

## Introduction

Structures de données  
linéaires

Structures de données  
arborescentes

Dictionnaire et  
table de hachage

Documenter  
un projet

Le but du cours est de savoir manipuler un ensemble de données selon un contexte imposé (par le problème ou par l'algorithme résolvant ce problème).

Pour un problème donné, on cherche à déterminer la meilleure façon de manipuler les données mises en jeu dans ce problème.

Pour cela on va

- *spécifier* = construire un modèle abstrait du réel et décrire ce modèle en vue d'assurer l'adéquation entre ce qui est produit et ce qu'on attend,

- *utiliser* une structure de données

*structure de données* = modèle qui décrit le comportement d'un ensemble d'informations muni de ses propriétés.

Une structure de données peut être vue de l'extérieur (modèle abstrait) et de l'intérieur (réalisation concrète).

## Introduction

Structures de données  
linéaires

Structures de données  
arborescentes

Dictionnaire et  
table de hachage

Documenter  
un projet

Le but du cours est de savoir manipuler un ensemble de données selon un contexte imposé (par le problème ou par l'algorithme résolvant ce problème).

Pour un problème donné, on cherche à déterminer la meilleure façon de manipuler les données mises en jeu dans ce problème.

Pour cela on va

- *spécifier* = construire un modèle abstrait du réel et décrire ce modèle en vue d'assurer l'adéquation entre ce qui est produit et ce qu'on attend,

- *utiliser* une structure de données

*structure de données* = modèle qui décrit le comportement d'un ensemble d'informations muni de ses propriétés.

Une structure de données peut être vue de l'extérieur (modèle abstrait) et de l'intérieur (réalisation concrète).

## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

Le but du cours est de savoir manipuler un ensemble de données selon un contexte imposé (par le problème ou par l'algorithme résolvant ce problème).

Pour un problème donné, on cherche à déterminer la meilleure façon de manipuler les données mises en jeu dans ce problème.

Pour cela on va

- *spécifier* = construire un modèle abstrait du réel et décrire ce modèle en vue d'assurer l'adéquation entre ce qui est produit et ce qu'on attend,

- *utiliser* une structure de données

*structure de données* = modèle qui décrit le comportement d'un ensemble d'informations muni de ses propriétés.

Une structure de données peut être vue de l'extérieur (modèle abstrait) et de l'intérieur (réalisation concrète).



## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

**Remarque** : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

**Remarque** : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

**Remarque** : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

Remarque : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

Remarque : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## étapes

- *spécification fonctionnelle* (ou *abstraite*) : on fait la liste des opérations (avec leurs définitions et restrictions) qui agiront sur les données. C'est le document de référence de l'utilisateur,
- *spécification opérationnelle* : elle consiste en deux étapes
  - description logique : on organise la structure de données en utilisant des structures informatiques classiques (par exemple des tableaux),
  - représentation physique : c'est l'implémentation complète faite dans un langage de programmation fixé.

**Remarque** : pour une même spécification fonctionnelle, il peut y avoir plusieurs spécifications opérationnelles dans le même langage.

## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

pour la spécification fonctionnelle (ou abstraite) on va définir un *type de données abstrait* avec :

- ① TYPE : les noms des types définis
- ② UTILISE : les types abstraits déjà définis utilisés
- ③ OPÉRATIONS : les opérations avec leur signature<sup>1</sup>
- ④ PRÉCONDITIONS : les restrictions éventuelles d'utilisation des opérations

---

1. la signature d'une opération est sa description syntactique

## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

pour la spécification fonctionnelle (ou abstraite) on va définir un *type de données abstrait* avec :

- ❶ TYPE : les noms des types définis
- ❷ UTILISE : les types abstraits déjà définis utilisés
- ❸ OPÉRATIONS : les opérations avec leur signature<sup>1</sup>
- ❹ PRÉCONDITIONS : les restrictions éventuelles d'utilisation des opérations

---

1. la signature d'une opération est sa description syntactique



## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

pour la spécification fonctionnelle (ou abstraite) on va définir un *type de données abstrait* avec :

- ❶ TYPE : les noms des types définis
- ❷ UTILISE : les types abstraits déjà définis utilisés
- ❸ OPÉRATIONS : les opérations avec leur signature<sup>1</sup>
- ❹ PRÉCONDITIONS : les restrictions éventuelles d'utilisation des opérations

---

1. la signature d'une opération est sa description syntactique

## Introduction

Structures de données linéaires

Structures de données arborescentes

Dictionnaire et table de hachage

Documenter un projet

pour la spécification fonctionnelle (ou abstraite) on va définir un *type de données abstrait* avec :

- ❶ TYPE : les noms des types définis
- ❷ UTILISE : les types abstraits déjà définis utilisés
- ❸ OPÉRATIONS : les opérations avec leur signature<sup>1</sup>
- ❹ PRÉCONDITIONS : les restrictions éventuelles d'utilisation des opérations

---

1. la signature d'une opération est sa description syntactique

Introduction

Structures de  
données  
linéaires

Généralités

Les tableaux

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## Structures de données linéaires

Une structure de données est linéaire lorsque les données sont en quelque sorte les unes derrière les autres : chaque donnée a une donnée successeur et une donnée prédécesseur exception faite éventuellement des données qui sont aux extrémités.

Une fois les données rangées, la structure sera amenée à évoluer par l'*ajout* de nouvelles données ou la *suppression* de données appartenant à la structure. Selon la façon de procéder à ces modifications, on distinguera plusieurs types de structures.

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Tableaux

Un tableau est une structure de données linéaire dans laquelle on a un accès direct à chaque donnée par son indice dans le tableau.

Les problèmes généralement posés sur les tableaux sont

- la recherche d'une valeur
- la recherche de la valeur la plus grande/petite
- le tri des valeurs dans l'ordre croissant
- le calcul de la valeur moyenne
- ...

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Tableaux

Un tableau est une structure de données linéaire dans laquelle on a un accès direct à chaque donnée par son indice dans le tableau.

Les problèmes généralement posés sur les tableaux sont

- la recherche d'une valeur
- la recherche de la valeur la plus grande/petite
- le tri des valeurs dans l'ordre croissant
- le calcul de la valeur moyenne
- ...

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Tableaux

Un tableau est une structure de données linéaire dans laquelle on a un accès direct à chaque donnée par son indice dans le tableau.

Les problèmes généralement posés sur les tableaux sont

- la recherche d'une valeur
- la recherche de la valeur la plus grande/petite
- le tri des valeurs dans l'ordre croissant
- le calcul de la valeur moyenne
- ...

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Tableaux

Un tableau est une structure de données linéaire dans laquelle on a un accès direct à chaque donnée par son indice dans le tableau.

Les problèmes généralement posés sur les tableaux sont

- la recherche d'une valeur
- la recherche de la valeur la plus grande/petite
- le tri des valeurs dans l'ordre croissant
- le calcul de la valeur moyenne
- . . .

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Tableaux

Un tableau est une structure de données linéaire dans laquelle on a un accès direct à chaque donnée par son indice dans le tableau.

Les problèmes généralement posés sur les tableaux sont

- la recherche d'une valeur
- la recherche de la valeur la plus grande/petite
- le tri des valeurs dans l'ordre croissant
- le calcul de la valeur moyenne
- ...



Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## 2-somme

**Problème : étant données  $n$  valeurs entières, trouver tous les couples de valeurs dont la somme est nulle.**

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## 2-somme

**Problème : étant données  $n$  valeurs entières, trouver tous les couples de valeurs dont la somme est nulle.**

algorithme : tester toutes les sommes, afficher les sommes nulles

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## 2-somme

**Problème : étant données  $n$  valeurs entières, trouver tous les couples de valeurs dont la somme est nulle.**

données : les  $n$  valeurs sont stockées dans un tableau d'entiers  $t$ .

algorithme : pour chaque indice  $k$  du tableau, calculer toutes les sommes  $t(k)+t(i)$  pour  $i$  variant dans les indices du tableau, les afficher si elles sont nulles

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## 2-somme

**Problème : étant données  $n$  valeurs entières, trouver tous les couples de valeurs dont la somme est nulle.**

données : les  $n$  valeurs sont stockées dans un tableau d'entiers  $t$ .

algorithme : pour chaque indice  $k$  du tableau, pour chaque indice  $i$  du tableau supérieur à  $k$ , calculer  $t(k)+t(i)$ , l'afficher si nulle

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## programme

```
public class DeuxSomme{
public static void afficherDeuxSomme(int [] t) {
    for (int k = 0; k < t.length; k++) {
        for (int i = k; i < t.length; i++) {
            if (t[k] + t[i] == 0) {
                System.out.println(t[k]+ " + " + t[i]);
            }
        }
    }
}

public static void main(String [] args) {
    . . .
}
```

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## temps d'exécution

Cette procédure exécute  $\frac{n(n-1)}{2}$  itérations donc la complexité est en  $O(n^2)$ .

Peut-on tester le temps d'exécution ? oui

Peut-on faire mieux ? oui

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## temps d'exécution

Cette procédure exécute  $\frac{n(n-1)}{2}$  itérations donc la complexité est en  $O(n^2)$ .

Peut-on tester le temps d'exécution ? oui

Peut-on faire mieux ? oui

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## temps d'exécution

Cette procédure exécute  $\frac{n(n-1)}{2}$  itérations donc la complexité est en  $O(n^2)$ .

Peut-on tester le temps d'exécution ? oui

Peut-on faire mieux ? oui



Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## temps d'exécution

Cette procédure exécute  $\frac{n(n-1)}{2}$  itérations donc la complexité est en  $O(n^2)$ .

Peut-on tester le temps d'exécution ? oui

Peut-on faire mieux ? oui

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## mesurer le temps d'exécution

```
public class Chronometre {  
  private final long debut = System.currentTimeMillis();  
  public double tempsEcoule() {  
    long maintenant = System.currentTimeMillis();  
    return (maintenant - debut) / 1000.0;}}
```

La méthode `currentTimeMillis()` de la classe `System` donne en milliseconde le temps écoulé depuis le 1er Janvier 1970 minuit jusqu'à l'invocation de la méthode.

Une instance de la classe `Chronometre` a donc l'attribut `debut` initialisé à cette durée au moment de l'appel au constructeur de la classe.

La méthode `tempsEcoule` mesure en seconde le temps écoulé entre l'appel au constructeur de la classe et l'invocation de la méthode.

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## mesurer le temps d'exécution

```
public static void main(String [] args) {  
    int [] t ;  
    // initialiser et remplir t  
    // utiliser la classe Random du paquetage java.util  
    Chronometre c = new Chronometre();  
    afficherDeuxSomme(t);  
    System.out.println  
    ("temps d'exécution " + c.tempsEcoule()+ " secondes");  
}
```

**exercice 1 :** : faire des tests sur des tableaux de dimension 10, 100, 1000, 2000, 4000, 8000.

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

Supposons le tableau classé par ordre croissant.

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

Supposons le tableau classé par ordre croissant.

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15



Introduction

Structures de données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

# Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

-2 et 2

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## Faire mieux ?

0	1	2	3	4	5	6	7	8	9	10	11
-7	-5	-4	-2	-1	2	3	6	7	8	11	15

-7 et 7

-2 et 2

arrêt car même signe

**exercice 2** : implémenter cette méthode

`afficherDeuxSommeRapide`



Introduction

Structures de données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## Tri par insertion

C'est le principe du joueur de carte :

on trie au fur et à mesure que l'on découvre une nouvelle valeur en l'insérant parmi celles déjà triées.

Par exemple : on a déjà

$i$	1	2	3	4	5	6	7	8	9	10
$t(i)$	-75	-54	15	34	56	74				

## Tri par insertion

C'est le principe du joueur de carte :

on trie au fur et à mesure que l'on découvre une nouvelle valeur en l'insérant parmi celles déjà triées.

Par exemple : on a déjà

$i$	1	2	3	4	5	6	7	8	9	10
$t(i)$	-75	-54	15	34	56	74				

et on donne la valeur 25 à insérer

$i$	1	2	3	4	5	6	7	8	9	10
$t(i)$	-75	-54	15	34	56	74				

## Tri par insertion

C'est le principe du joueur de carte :

on trie au fur et à mesure que l'on découvre une nouvelle valeur en l'insérant parmi celles déjà triées.

Par exemple : on a déjà

$i$	1	2	3	4	5	6	7	8	9	10
$t(i)$	-75	-54	15	34	56	74				

On décale les trois valeurs supérieures à 25 d'un rang vers la droite puis on place 25

$i$	1	2	3	4	5	6	7	8	9	10
$t(i)$	-75	-54	15	25	34	56	74			

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

En pratique, le tableau est déjà rempli. Donc on commence par insérer la deuxième valeur dans la partie gauche du tableau réduite à la première valeur, puis la troisième valeur dans la partie gauche du tableau composée des deux premières valeurs déjà triées et ainsi de suite jusqu'à insérer la dernière valeur.

L'insertion peut se réaliser de façons

- on décale d'un rang vers la droite toutes les valeurs de rang inférieur qui sont supérieures à la valeur à insérer
- on recule la valeur à insérer par échange de la valeur avec sa voisine de gauche jusqu'à ce sa voisine lui soit inférieure

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

En pratique, le tableau est déjà rempli. Donc on commence par insérer la deuxième valeur dans la partie gauche du tableau réduite à la première valeur, puis la troisième valeur dans la partie gauche du tableau composée des deux premières valeurs déjà triées et ainsi de suite jusqu'à insérer la dernière valeur.

L'insertion peut se réaliser de façons

- on décale d'un rang vers la droite toutes les valeurs de rang inférieur qui sont supérieures à la valeur à insérer
- on recule la valeur à insérer par échange de la valeur avec sa voisine de gauche jusqu'à ce sa voisine lui soit inférieure

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

En pratique, le tableau est déjà rempli. Donc on commence par insérer la deuxième valeur dans la partie gauche du tableau réduite à la première valeur, puis la troisième valeur dans la partie gauche du tableau composée des deux premières valeurs déjà triées et ainsi de suite jusqu'à insérer la dernière valeur.

L'insertion peut se réaliser de façons

- on décale d'un rang vers la droite toutes les valeurs de rang inférieur qui sont supérieures à la valeur à insérer
- on recule la valeur à insérer par échange de la valeur avec sa voisine de gauche jusqu'à ce sa voisine lui soit inférieure

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

En pratique, le tableau est déjà rempli. Donc on commence par insérer la deuxième valeur dans la partie gauche du tableau réduite à la première valeur, puis la troisième valeur dans la partie gauche du tableau composée des deux premières valeurs déjà triées et ainsi de suite jusqu'à insérer la dernière valeur.

L'insertion peut se réaliser de façons

- on décale d'un rang vers la droite toutes les valeurs de rang inférieur qui sont supérieures à la valeur à insérer
- on recule la valeur à insérer par échange de la valeur avec sa voisine de gauche jusqu'à ce sa voisine lui soit inférieure

Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## Tris de tableau efficaces

Les tris simples ont une complexité en  $O(n^2)$ . Donc la méthode précédente pour afficher toutes les sommes nulles de deux nombres du tableau n'apporte rien si on ne peut pas améliorer les tris.

Les tris rapide, fusion ou par tas ont une complexité en moyenne de  $O(n \ln(n))$ .



Introduction

Structures de  
données  
linéaires

Généralités

**Les tableaux**

Autres structures  
linéaires : un  
exemple

Les piles

Exercices

les files

Exercices

Les listes

Exercices

Structures de  
données  
arborescentes

Dictionnaire et  
table de  
hachage

Documenter  
un projet

## Tri rapide

**exercice 3** : Ecrire les méthodes `pivoter` puis `triRapide`

**exercice 4** : tester le temps d'exécution de `tri rapide` +  
`afficherDeuxSommeRapide`.