

Object Oriented programming with java

Algorithme de tri et de recherche

Clervil Wilky

École Supérieur d'Infotronique d'Haïti

Port-au-Prince, Haïti

27 Septembre 2021

Table des matières

Introduction	2
Recherche par saut	2
Recherche exponentielle	3
Le tri rapide	4
Tri par insertion	5
Conclusion.....	5
`Références.....	6

Introduction

Dans ce document, nous allons expliquer deux algorithmes de tri et deux algorithmes de recherche que nous avons implémentée. Les deux algorithmes de recherche que nous allons expliquer sont le recherche par saut et le recherche exponentielle.

Recherche par saut

Pour utiliser l'algorithme de recherche par saut, le tableau doit être trié. L'idée de base est de faire moins de tour de boucle que pour la recherche linéaire en sautant certains éléments du tableau au lieu de tester tous les éléments.

Pour effectuer une recherche par saut, il faut :

1. Choisir la taille de bloc optimale à sauter.
2. Choisir deux indices, un indice servant de borne supérieur et un autre servant de borne inférieur. nous nommerons ces deux indices, indice1 pour la borne inferieur et indice2 pour la borne supérieur.
3. Parcourir tous les éléments du tableau, jusqu'à ce qu'on trouve un élément supérieur ou égal à celui qui est recherché.
4. Arrêter le parcours si aucun élément n'est retrouvé
5. Stocker l'indice de l'élément supérieur ou égal à celui qui est recherché dans l'indice2, si ce dernier a été retrouvé.
6. L'indice1 servira de variable pour stocker l'indice de l'élément qui se trouve d'un saut avant l'indice de borne supérieur.
7. Faire une recherche linéaire dans l'intervalle comprise entre l'indice1 et l'indice2.
8. Si l'élément recherché ne se trouve pas sur cet intervalle, c'est qu'il ne se trouve pas dans le tableau tout simplement.

Pour trouver la taille de bloc optimale, il suffit de prendre la racine carré de la taille du tableau.

Recherche exponentielle

Pour utiliser l'algorithme de recherche exponentielle, le tableau doit être trié, Le nom de cet algorithme vient de la façon dont il recherche un élément. La recherche exponentielle a deux étapes :

1. Trouver l'intervalle dont l'élément est présent
2. Faites une recherche dichotomique dans l'intervalle dont l'élément est présent

Pour trouver l'intervalle dont l'élément est présent il faut :

1. Choisir un compteur, dans cet algorithme nous le nommerons i .
2. Initialiser i à 1.
3. Parcourir le tableau en comparant les éléments se trouvant à l'indice i avec celui qui est recherché.
4. A chaque tour de boucle il faut multiplier i par 2 et s'arrêter une fois qu'un élément plus grand ou égal à celui recherché a été retrouvé ou si i est plus grand ou égal à la taille du tableau.
5. Stocker le minimum entre i et la taille du tableau moins un dans une variable, pour cet algorithme nous nommerons ce variable min .
6. Effectuer une recherche dichotomique sur l'intervalle comprise entre $i/2$ et min .

Pour faire la recherche dichotomique il faut :

- 1) Séparer le tableau en deux parties égales
- 2) Comparer la valeur recherchée avec celle située au milieu du tableau
- 3) Continuer la recherche dans un seul des deux sous tableaux, si la valeur recherchée ne se trouve pas au milieu
- 4) Arrêter la recherche si le tableau ne peut plus être divisé ou si la valeur a été retrouvée.

Le tri rapide

` Le tri rapide est basé sur la récursivité et la dichotomie. Pour faire un tri rapide, il faut :

1. Choisir un élément appelé pivot
2. Placer les éléments plus grands que le pivot à sa droite et ceux plus petits que le pivot à sa gauche, de telle sorte que le pivot se trouve à une position fixe dans le tableau et n'a plus besoin d'être déplacé.
3. Diviser le tableau en deux parties, la première partie ayant les éléments se trouvant à droite du pivot et la deuxième ayant les éléments se trouvant à gauche du pivot.
4. Recommence à trier la partie du tableau situant à gauche du pivot et la partie du tableau situant à droite du pivot.

Pour placer le pivot, il faut :

1. Utiliser deux variables qu'on nommera pour dans ce document `indiceGauche` initialisé avec la position de l'élément situe juste après le pivot et `indiceDroite` initialisé avec la position de l'élément situe à la fin du tableau.
2. Parcourir le tableau en partant de la gauche vers la droite, en incrémentant `indiceGauche`. Il faut s'arrêter lorsqu'on trouve un élément plus grand que le pivot.
3. Parcourir le tableau en partant de la droite vers la gauche, en décrémentant `indiceDroite`. Il faut s'arrêter lorsqu'on trouve un élément plus petit que le pivot
4. Echanger la position des éléments trouvés en parcourant le tableau avec les variables `indiceGauche` et `indiceDroite`.
5. Continuer à parcourir le tableau jusqu'à ce que `indiceGauche` et `indiceDroite` désigne le même élément.

Tri par insertion

Le tri par insertion permet de trier un tableau. Cette une méthode de tri simple à comprendre et facile à mettre en œuvre.

Pour faire un tri par insertion il faut :

1. Utiliser une boucle principale afin de parcourir tous les éléments du tableau, en commençant à partir de l'élément situé à la deuxième position.
2. Sauvegarder l'élément à insérer dans une variable, L'élément à insérer étant l'élément de la boucle principale.
3. Utiliser une autre boucle, pour parcourir le tableau, en partant de l'élément à insérer jusqu'à la fin du tableau.
4. Décaler tous les éléments plus grands que l'élément à insérer vers la droite.
5. Inscrire l'élément à insérer dès qu'on trouve un élément plus petit ou qu'on arrive à la fin du tableau.

Voici un exemple très simple du tri par insertion

Tableau initial

4	3	2	7
---	---	---	---

Placer 3 avant 4

3	4	2	7
---	---	---	---

Placer 2 avant 3,4

2	3	4	7
---	---	---	---

7 est déjà à sa place par rapport 2,3,4

Conclusion

Comme nous l'avons mentionné au début, on a expliqué deux algorithmes de tri et de recherche. A noter qu'il y'a d'autres algorithmes de tri et chacun d'eux présentent des avantages et des inconvénients.

^Références

Christophe Dabancourt, Apprendre à programmer (algorithmes et conceptions d'objet), 2^{eme} édition, Juillet 2008, 313 pages