

**Prosit Aller**

Modularisation, tris et recherches

**SOMMAIL**

Mots de clés ………………………………………………………...................1

Contexte……………………………………………………..………………………..1

Contrainte ………………………………….……………………………….………1

Besoins…………………………………………….……………………………….….1

Problématiques……………………………………………………………….……1

Généralisation………………………………………………………………........1

Hypothèses……………………………………………………………………........1

Pistes………………………………………………………………….………..……….1

**Plan d’action**

**Etude……………………………………………………………………….I**

Définition…………………………………………………….…………………….…..1

Ressources……………………………………………………………………….…….2

Tableau en C…………………………………………………………………………..3

La recherche dichotomique ……………………………………………..…….4

Les types de tri…………………………………………………………………….…5

**REALISATION……………………………………………………………………..…II**

Comprendre la fonction du programme (logigramme)…………..1

Réalisation du programme……………………………………………………..2

**CONCLUSION……………………………………………………………………….III**

Interprétation des résultats…………………………………………………….1

 Validation des hypothèses …………………………………………………….2

 Réponses aux problématiques ……………………………………………….3

**Mots clés :**

Radar

Recherche dichotomique

Les 4 types d’infraction

Tri par sélection

Système

Tri à bulles

Tableau

Tri fusion récursive

Structure

**Contexte :**

Ils se sont fait flasher et veulent savoir combien ça va leur coûter.

**Analyse des besoins, contraintes :**

Tableau

Langage C

Enumération

Tri

Fonctions

Recherche dichotomique

**Problématiques :**

Comment fonctionne le système pour calculer les amandes ?

**Généralisation :**

Tri de tableau en C

**Hypothèses :**

Les radars contiennent un programme : oui

Les tableaux permettent d’organiser des séries de valeur : faux

On utilise des fonctions pour organiser des valeurs dans un tableau : oui

Il existe une fonction tableau en C : oui

La recherche dichotomique permet de trouver un élément trié : oui

Il existe d’autres types de tri : oui

**Plan d’action :**

**I. ETUDE**

1. Définition des mots clés

2. A voir/Personnel (Ex : Etude des ressources, etc…)

3. Tableau en C

4. La recherche dichotomique

5. Les types de tri

**II. REALISATION**

1. Comprendre la fonction du programme (logigramme)

2. Réalisation du programme

**III. CONCLUSION**

1. Interprétation des résultats

2. Validation des hypothèses

3. Réponses aux problématiques

**Plan d’action :**

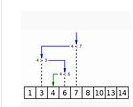
**I. ETUDE**

**1. Définition des mots clés**

**Radar** est un système qui utilise les ondes électromagnétiques pour détecter la présence et déterminer la position ainsi que la vitesse.

**Recherche dichotomique** est un algorithme de recherche pour trouver la position d'un élément dans un tableau trié. Le principe est le suivant : comparer l'élément avec la valeur de la case au milieu du tableau ; si les valeurs sont égales, l…

**Exemple**



**Les 4 types d’infraction**

**Tri par sélection** est un algorithme de tri par comparaison. Cet algorithme est simple, mais considéré comme inefficace.

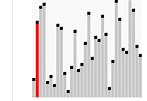
**Exemple**



**Système** est un ensemble d'éléments interagissant entre eux selon certains principes ou règles.

**Tri à bulles** est un algorithme de tri. Il consiste à comparer répétitivement les éléments consécutifs d'un tableau.

**Exemple**



**Tableau** est une suite de variables de même type, situées dans un espace contigu en mémoire.

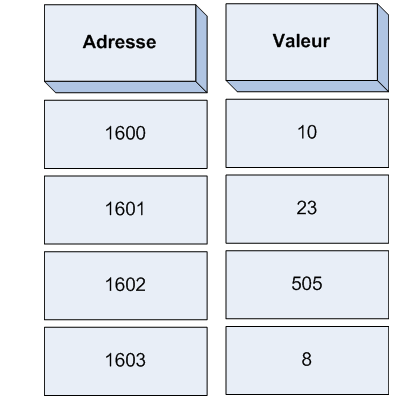
**Tri fusion récursive** est un algorithme de tri par comparaison stable. Sa complexité temporelle pour une entrée de taille n est de l'ordre de n log n.

**Structure** est la façon dont les éléments participants d'un système sont organisés entre eux**.**

**2- Etude des ressources**

**3. Tableau en C**

Un tableau a une dimension bien précise. Il peut occuper 2, 3, 10, 150, 2 500 cases. La fig. suivante est un schéma d'un tableau de 4 cases en mémoire qui commence à l'adresse 1600.



Lorsqu’on veut créer un tableau de 4 cases en mémoire, votre programme demande à l'OS la permission d'utiliser 4 cases en mémoire. Ces 4 cases doivent être contiguës, c'est-à-dire les unes à la suite des autres. Comme vous le voyez, les adresses se suivent : 1600, 1601, 1602, 1603. Il n'y a pas de « trou » au milieu.

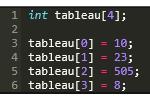
Chaque case du tableau contient un nombre du même type. Si le tableau est de type **Int**, alors chaque case du tableau contiendra un **Int**. On ne peut pas faire de tableau contenant à la fois des **Int** et des **double** par exemple.

-Lorsqu'un tableau est créé, il prend un espace contigu en mémoire : les cases sont les unes à la suite des autres.

-Toutes les cases d'un tableau sont du même type. Ainsi, un tableau de Int contiendra uniquement des Int, et pas autre chose.

Le tableau de 4 **Int** a donc les indices 0, 1, 2 et 3. Il n'y a pas d'indice 4 dans un tableau de 4 cases.

**Exemple**



**4. La recherche dichotomique**

Une manière efficace et rapide de rechercher un élément dans une structure de données triée (pour nous, ce sera un tableau trié).

Avec une **recherche dichotomique**, on ne va pas parcourir le tableau de façon linéaire, c'est-à-dire du début jusqu'à la fin. Plutôt que cette méthode, on va utiliser un critère de comparaison qui sera en adéquation avec le critère que l'on a utilisé pour trier le tableau. L'idée ici est d'employer le fait que le tableau soit trié avec des valeurs classées par ordre croissant.

**Exemple**



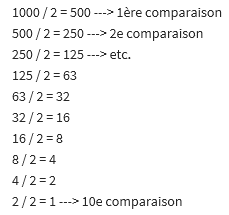
On a ici un tableau rempli de "gauche à droite" par des entiers avec 12 qui est inférieur à 14 qui est lui-même inférieur à 20, etc. L'idée est donc au départ (= au premier parcours de la boucle de recherche) de regarder au milieu du tableau. Ainsi, là, on sait que l'on a 8 valeurs dans le tableau, il suffit de diviser ce nombre par 2 pour avoir l'indice du milieu : 4. Ici tab[4] correspond donc à la valeur 26. On compare donc tab[4] à la valeur recherchée.

Si celle-ci est supérieure à 26, on ne va rechercher notre valeur que dans la partie "supérieure" du tableau, c'est-à-dire les valeurs des cases qui ont des indices compris entre 4 et 8. Et inversement, si celle-ci est inférieure à 26, on ne recherchera notre valeur que dans la partie "inférieure" du tableau, avec des indices compris entre 0 et 4.

Évidemment, si la valeur recherchée est dans la case dont on a obtenu l'indice(4), on arrêtera la recherche.

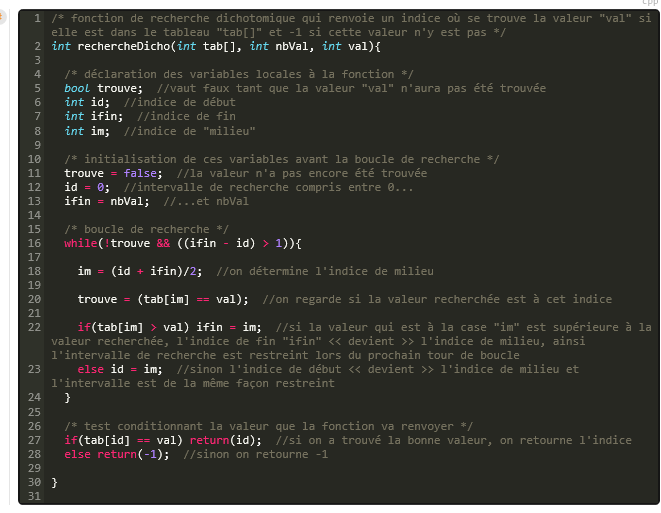
Une autre méthode si vous avez 1000 valeurs dans votre tableau et Si l'intervalle se réduit à chaque fois par deux.

On a donc



10 comparaisons maximum, et en sachant qu'une comparaison = 1 tour de boucle, c'est beaucoup plus efficace que si on recherchait la valeur dans chaque case du tableau et où l'on devrait faire 1000 comparaisons et donc 1000 tours de boucle si la valeur était dans la dernière case.

**Le code de la fonction de recherche dichotomique**



**5. Les types de tri**

Tri à bulles, tri par sélection, tri par insertion, tri par tas, tri par base, tri par paquets, tri rapide, tri fusion, tri comptage,

**II. REALISATION**

**1. Comprendre la fonction du programme (logigramme)**

**2. Réalisation du programme**

**III. CONCLUSION**

**1. Interprétation des résultats**

**2. Validation des hypothèses**

**3. Réponses aux problématiques**