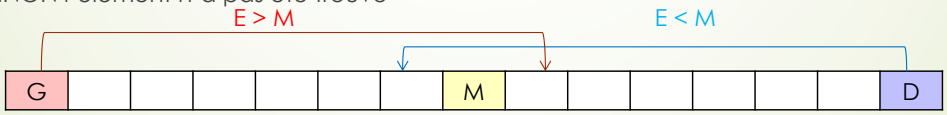
7.8 Recherche binaire

31

Recherche binaire

- La recherche séquentielle fonctionne, mais elle n'est pas très efficace dans le cas d'un tableau trié. En effet, il existe d'autres méthodes de recherche, tel que la recherche binaire, qui sont beaucoup plus rapides pour de grands tableaux.
  - Par exemple, pour un tableau de 1 000 000 d'éléments, il faut faire 500 000 boucles en moyenne dans le cas de la recherche séquentielle, alors que 20 boucles ($\log_2 N$) sont suffisantes à l'aide d'une recherche binaire.
- La recherche binaire implémente la stratégie *diviser pour régner*. À chaque étape, on réduit de moitié le nombre de candidats en déplaçant les bornes de la recherche.

Initialiser les bornes de gauche et de droite aux extrémités du tableau
Initialiser le milieu avec une division entière: (gauche + droite) / 2
TANT QUE l'élément n'est pas trouvé au milieu et que les bornes ne se croisent pas FAIRE
    SI l'élément < milieu ALORS déplacer la borne de droite à gauche du milieu
    SI l'élément > milieu ALORS déplacer la borne de gauche à droite du milieu
    Mettre à jour le milieu
SI les bornes ne se croisent pas ALORS l'élément cherché est à la position du milieu
SINON l'élément n'a pas été trouvé



Farida Ait-Ammar et Nicolas Leduc

32

Recherche binaire

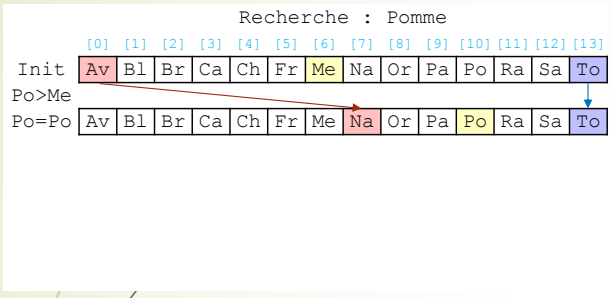
```
public static void main(String[] args) {
     Scanner clavier = new Scanner(System.in);
     String[] nomProd = {"Avocat","Bleuet","Brocoli","Carotte","Chou","Fraise","Melon",
                         "Navet","Orange","Patate","Pomme","Raisin","Salade","Tomate",""};
     int[] quant = {1, 4, 2, 5, 1, 3, 1, 1, 5, 4, 7, 2, 1, 6, -1};
     int nbElem = 14;
     String prodCherche;

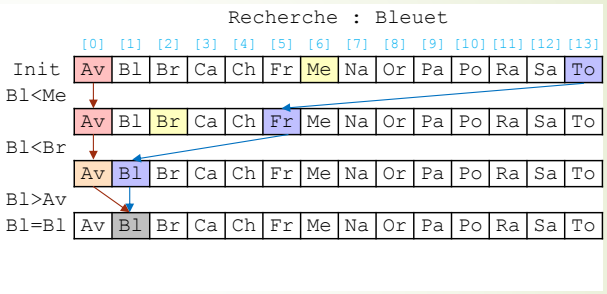
     System.out.print("Entrez le nom d'un produit : ");
     prodCherche = clavier.next();
     int posGa = 0;
     int posDr = nbElem - 1;
     int posMi = (posGa + posDr) / 2;      // Division entière pour trouver le milieu
     System.out.println(posGa + " : " + posMi + " : " + posDr);              // Affichage des bornes
     // On s'arrête lorsque les bornes se croisent ou lorsqu'on a trouvé le produit au milieu
     while (posGa <= posDr && !prodCherche.equals(nomProd[posMi])) {
          if (prodCherche.compareTo(nomProd[posMi]) < 0) {
               posDr = posMi - 1;          // Si on est à gauche du milieu, on déplace la borne de droite
          } else {
               posGa = posMi + 1;          // Si on est à droite du milieu, on déplace la borne de gauche
          }
          posMi = (posGa + posDr) / 2;     // Mise à jour du milieu
          System.out.println(posGa + " : " + posMi + " : " + posDr);
     }
     if (posGa <= posDr) {                 // Si les bornes ne se croisent pas, on a trouvé
          System.out.println("Produit trouvé: " + nomProd[posMi] + " (" + quant[posMi] + ")");
     } else {
          System.out.println("Aucun produit trouvé");
     }
}
```

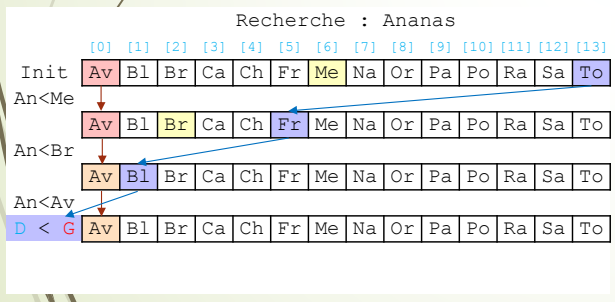
Farida Ait-Ammar et Nicolas Leduc

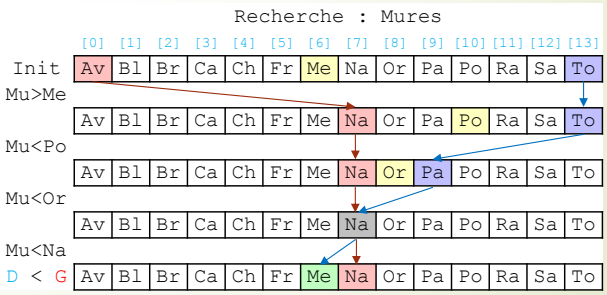
33

Recherche binaire









Farida Ait-Ammar et Nicolas Leduc