

# Recherche par dichotomie

La recherche de la présence d'un élément dans une liste a déjà été vue au TP 1. Nous proposons dans ce TP une autre approche de cette recherche, nous l'appliquerons à une situation concrète puis nous ferons une analyse de cette nouvelle méthode.

## . DÉCOUVERTE DE LA DICHOTOMIE

Regarder la [vidéo suivante](#)

### . EXERCICE 1 : POINT DE VUE DU COMMERÇANT

- Au début de la discussion avec le client, quel protocole le fromager propose-t-il au client ?**
  - Tourner le couteau sur le fromage pour trouver la bonne position,
  - Descendre le couteau le long du fromage pour trouver la bonne position,
  - Diviser le fromage en deux parties égales.
- Quelle phrase illustre le mieux la méthode du fromager ?**
  - Il déplace le couteau de manière aléatoire jusqu'à que le client identifie la bonne position ?
  - Il réduit progressivement la part proposée jusqu'à que le client identifie la part qu'il souhaite.
  - Il réduit la part dans laquelle il cherche la position du couteau, correspondant à la coupe souhaitée par le client.
- Combien de positions précédemment occupées par la lame, le fromager doit-il avoir en mémoire à un instant donné (en plus de la position actuelle) ?**

1                    2                    3                    toutes

### . EXERCICE 2 : POINT DE VUE DU PROGRAMMEUR

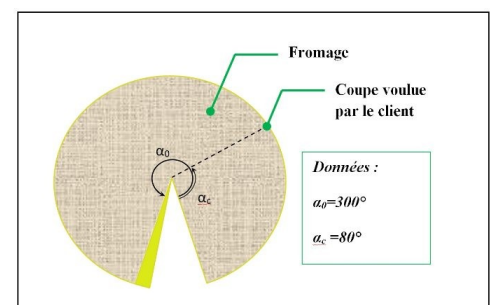
- Pour mémoriser la position de la lame, le programmeur réalisera :**
  - une affectation de variable
  - un test
  - un affichage de valeur
- La discussion correspond à**
  - une boucle bornée
  - une boucle non bornée
  - une suite d'instruction sans boucle.
- L'écart entre deux positions successives réduit à chaque fois ...**
  - d'une valeur constante
  - de moitié (environ)
  - d'une valeur aléatoire

- On considère la situation décrite par l'image ci-contre.**

On définit la position de la lame par un angle  $\beta$ .

Quelles seraient les deux premières valeurs de  $\beta$  ?

Quelle sera la valeur suivante de  $\beta$  ?



- On considère maintenant que l'angle  $\alpha_0$  est égal à 280 et que l'angle voulu par le client est inconnu.**
  - Quels seront les 4 premiers angles  $\beta$  repérant les positions occupées par la lame si les indications successives du client pendant la discussion sont : "plus", "moins", "moins", ?
  - Quels seront les 4 premiers angles  $\beta$  repérant les positions occupées par la lame si les indications successives du client pendant la discussion sont : "moins", "plus", "plus" ?

## . APPLICATION DANS UN MAGASIN DE SURF

On suppose que vous êtes gérant d'un magasin de surf. Un client veut acheter une planche d'une taille précise. Il est pressé, heureusement, les planches sont rangées par ordre de tailles croissantes. Nous allons donc utiliser la recherche par dichotomie vue dans la partie 1 de ce TP.



### . EXERCICE 3 : IMPLÉMENTATION EN PYTHON DE LA RECHERCHE PAR DICHOTOMIE

1. En Python, créer la fonction `recherche_dichotomie( planches : list, valeur : int)` et dont le code implémente l'algorithme ci-dessous.

```
Initialiser la variable i_min à 0
Initialiser la variable i_max à len( planches ) - 1
Tant que i_min est inférieur ou égale i_max
    index prend pour valeur la partie entière de (i_min + i_max)/2
    Si planches[index] est égale à valeur alors renvoyer index
    Sinon si planches[index] < valeur, i_min devient index + 1
    Sinon i_max devient index - 1
Renvoyer None
```

2. Écrire le docstring de cette fonction

### . EXERCICE 4 : APPLICATION À LA RECHERCHE D'UNE PLANCHE DE SURF

1. Créer une liste de 60 planches de surf. La plus première mesure 100 cm. Chaque planche mesure entre 1 et 10 cm de plus que la précédente.
2. Choisir une valeur au hasard comprise entre 100 et la dernière valeur de la liste.
3. Faire un appel à la fonction `recherche_dichotomie` pour identifier l'indice de la valeur choisie dans la liste des planches de surf.
4. Écrire les lignes permettant de vérifier que l'indice récupéré est correct et d'afficher un message d'erreurs en cas de problème.
5. Exécuter plusieurs fois le programme pour observer les éventuels messages d'erreurs et faire les corrections nécessaires.

## . ANALYSE DE L'ALGORITHME

On rappelle que la recherche par dichotomie s'applique uniquement sur des listes triées.

### . EXERCICE 5 : ÉTUDE DE LA COMPLEXITÉ.

1. Préciser les cas les plus défavorables ? (élément recherché au milieu, au début, à la fin, ...)
2. Parmi les complexités suivantes, laquelle vous semble être celle de l'algorithme de recherche par dichotomie :  $\log_2(n)$        $n \cdot \log_2(n)$        $n$   
Proposer une expérimentation ou une démonstration permettant de justifier votre réponse.

### . EXERCICE 6 : RECHERCHE D'UN VARIANT DE BOUCLE

Pour assurer la terminaison de notre boucle nous allons chercher **un variant** (grandeur bornée d'un côté variant à chaque tour de boucle vers sa borne)

1. **Choix du variant:** Parmi les grandeurs suivantes quelle(s) est (sont) celle(s) qui varie(nt) de manière strictes à chaque tour de la boucle *while* ?  
a)  $i_{\min}$       b)  $i_{\max}$       c) la longueur de la liste      d)  $i_{\max} - i_{\min}$       e)  $i_{\max} / i_{\min}$
2. **Choix de la borne inférieur du variant :** Parmi les conditions suivantes, laquelle détermine l'arrêt de la boucle ?  
a)  $\text{variant} > 1$       b)  $\text{variant} < 0$       c)  $\text{variant} = 0$
3. **L'existence d'un variant borné assure-t-il la validité de l'algorithme ?**  
a) OUI      b) NON      c) Cela dépend des cas.