TD paradigme diviser pour régner :

Recherche du majoritaire

Majoritaire :

1. Définition :

Un élément x est dit majoritaire dans un ensemble E de n éléments, si et seulement si le nombre d'occurrences de x dans E est strictement supérieur à n/2.

2. Exemple :

18 filles dans une classe de 34 élèves.

3. Hypothèse :

Pour simplifier on prendra n pair.

4. Propriétés :

Il y a au plus un élément majoritaire (s'en convaincre !).
Si on sépare E = E1 ∪ E2 avec E1 et E2 de taille n/2
si x est majoritaire dans E, alors x est majoritaire dans E1 ou dans E2.
si x est majoritaire dans E1 et dans E2, alors il est majoritaire dans E. (le démontrer).

5. Algorithme naïf :

Ecrire un code python de recherche du majoritaire et mesurer son efficacité pour des valeurs de n croissantes.

6. Diviser pour régner :

En explorant récursivement les deux moitiés de tableau, écrivez une fonction ***majoritaire(i,j)*** qui cherche s'il existe un élément majoritaire dans E[i..j].

Elle retourne :
(-, 0) s'il n'y a pas de majoritaire dans E[i..j].
(x, c) si x est majoritaire avec un nombre d'occurrences égal à c.

On utilise une fonction occ(x, i, j) qui compte le nombre d'occurrences de x dans E[i..j] en comparant x à chacun des éléments.

```
fonction majoritaire(i, j)
     si i=j alors retourner (E[i], 1)
     sinon
        milieu <- floor((i+j)/2)
        taille <- j-i+1
        (x, cx) <- majoritaire(i, milieu)
        (y, cy) <- majoritaire(milieu+1, j)
        si x=y alors retourner (x, cx+cy)
        si cx <> 0 alors cx <- cx + occ(x, milieu+1, j)
        si cy <> 0 alors cy <- cy + occ(y, i, milieu)
        si cx > taille/2 alors retourner (x, cx)
        sinon si  cy > taille/2 alors retourner (y, cy)
              sinon retourner (-, 0)
```

Ecrire le code python de recherche du majoritaire avec ce paradigme et mesurer son efficacité pour des valeurs de n croissantes.

Comparer l’efficacité de vos 2 programmes.