FEUILLE DE TRAVAUX DIRIGÉS 1**Exercice 1. Recherche dichotomique pour approximer la racine carrée :**

Étant donné un nombre $x \geq 0$, on cherche à approximer sa racine carrée $\sqrt{x}$ avec une précision donnée $\varepsilon > 0$.

1. Expliquer le principe de la recherche dichotomique pour résoudre ce problème.
2. Appliquer cet algorithme pour $x = 10$ avec $\varepsilon = 0.01$.
3. Écrire une fonction `racineCarree(x, ε)` en pseudo-code.
4. Quelle est la complexité temporelle de cet algorithme en fonction de ε ?

Exercice 2. Principe du tri à bulles :

On parcourt le tableau en inversant un élément avec son successeur s'il lui est supérieur. Après un premier parcours, le tableau est partiellement ordonné, mais il reste des couples d'éléments inversés. On recommencera donc le parcours jusqu'à ce que l'on ne rencontre plus de permutations.

On considère un tableau A de taille n indexé de 0 à $n - 1$.

1. Dérouler cet algorithme sur l'exemple $[5, 9, 10, 7]$.
2. Écrire un algorithme `parcours()` qui effectue un parcours de gauche à droite.
3. Écrire un algorithme `triabulles()` qui effectue le tri à bulles du tableau.
4. Calculer la complexité de ce tri.

Exercice 3. Principe du tri fusion :

Le tri fusion est un algorithme diviser-pour-régner.

- DIVISER : On partage le tableau initial en deux sous-tableaux de taille égale à 1 près (taille initiale impaire).
- RÉGNER : On poursuit récursivement le partage jusqu'à des tableaux élémentaires de taille 1.
- COMBINER : On rassemble deux tableaux triés pour n'en obtenir qu'un seul, lui aussi trié.

1. Appliquer cet algorithme sur l'exemple $[3, 4, 6, 2, 5, 1, 8, 7]$. Même question pour $[5, 7, 9, 1, 2]$.
2. Écrire une fonction `combiner(arr1, arr2)` qui prend en paramètre deux tableaux dont les éléments sont triés dans l'ordre croissant et qui renvoie un tableau trié par ordre croissant dont les éléments sont ceux des deux tableaux précédents.
3. Écrire une fonction `triFusion(arr)` en pseudo-code.
4. Quelle est la complexité moyenne de cet algorithme ?

Exercice 4. Bonus issu de LeetCode : Plateforme en ligne de programmation pour se préparer aux entretiens :

PROBLÈME 852 : SOMMET D'UN TABLEAU DE MONTAGNE

On vous donne un tableau de montagne entier `arr` de longueur `n`, où les valeurs augmentent jusqu'à atteindre un élément sommet, puis diminuent.

Écrivez une fonction qui retourne l'indice de l'élément sommet.

Définition d'un tableau de montagne. Un tableau de montagne est défini comme suit :

- La longueur du tableau est d'au moins 3.
- Il existe un indice i ($0 < i < n - 1$) tel que :
 - `arr[0] < arr[1] < ... < arr[i]` (les valeurs augmentent strictement jusqu'à l'indice i).
 - `arr[i] > arr[i+1] > ... > arr[n-1]` (les valeurs diminuent strictement après l'indice i).

Exemples.

- **Exemple 1 :**
Entrée : `arr = [0,1,0]`
Sortie : 1
- **Exemple 2 :**
Entrée : `arr = [0,2,1,0]`
Sortie : 1
- **Exemple 3 :**
Entrée : `arr = [0,10,5,2]`
Sortie : 1

Contraintes.

- $3 \leq \text{arr.length} \leq 10^4$
- $0 \leq \text{arr}[i] \leq 10^6$
- Les éléments de `arr` sont distincts.
- `arr` est garanti d'être un tableau de montagne.