TD1 : Tableaux dynamiques

Objectif : Écrire une classe `TableauTrie` qui conserve des entiers dans un tableau **en ordre croissant**. Comprendre le coût des opérations (taille, insertion triée, affichage, accès, retrait, modification, recherche linéaire et dichotomique).

Le squelette est fourni dans `TableauTrie.java` (sur ARCHE).

Construction de l'objet

- Constructeur par défaut : crée un tableau vide.
- Second constructeur `TableauTrie(int n)` : remplit le tableau avec `n` entiers puis **trie** le tableau.

Questions :

Q1. taille : retourner le nombre d'éléments.
Q2. push : insérer un entier en conservant l'ordre croissant.
Q3. afficher : afficher le contenu du tableau.
Q4. valeur(ind) : retourner la valeur à l'indice `ind`. Lever une exception si l'indice est invalide.
Q5. retirer(ind) : retirer l'élément à `ind` et retourner sa valeur. Le tableau doit rester trié. Lever une exception si l'indice est invalide.
Q6. modification(ind, v) : remplacer la valeur à `ind` par `v`, puis restaurer l'ordre croissant. Lever une exception si l'indice est invalide.
Q7. recherche(x) (linéaire) : retourner l'indice si présent, `-1` sinon.
Q8. recherche dichotomique : retourner l'indice si présent, `-1` sinon.

Rappel : recherche dichotomique

Sur un tableau **trié** :

- comparer la cible `e` à l'élément du milieu ;
- si égal → renvoyer l'indice ;
- si `e` plus petit → chercher dans la moitié gauche ;
- si `e` plus grand → chercher dans la moitié droite ;
- s'arrêter si l'intervalle devient vide.

Indications pratiques

- Milieu : `int milieu = (ind1 + ind2) / 2;` (intervalle semi-ouvert conseillé : $[ind1, ind2)$).
- Vérifier systématiquement les indices ($0 \leq$ `ind` $<$ `taille`) et lever une exception si invalide.