

Quick 1 – 6 octobre 2023 – durée 1 h

Les documents et les téléphones (incluant smartphone, tablettes,... tout ce qui contient une interface réseau) interdits. Les calculatrices sont autorisées.

Seuls les dictionnaires pour les personnes de langue étrangère sont autorisés.

Le barème est indicatif.

Vous êtes vivement encouragés à illustrer vos recherches et vos réponses par des schémas.

Question 1 : Écriture d'algorithme et complexité (6 points)

On considère l'algorithme suivant :

CHOIX(T)

Données : Un tableau T de n éléments comparables

Résultat : ???

```
1  $i := 0$ 
2  $j := n - 1$ 
3 while  $i < j$ 
4   if  $T[i] < T[j]$ 
5      $i := i + 1$ 
6   else
7      $j := j - 1$ 
7 return  $i$ 
```

- (2 pts) Décrivez précisément ce que représente le résultat renvoyé par cet algorithme.
- (1 pts) Quel est le coût au pire de cet algorithme, en nombre de comparaisons d'éléments de T ?
Donnez une réponse précise, pas seulement un ordre de grandeur.
- (1 pts) Quel est le coût au mieux de cet algorithme, en nombre de comparaisons d'éléments de T ?
Donnez une réponse précise, pas seulement un ordre de grandeur.
- (2 pts) On suppose maintenant que le tableau T est trié au préalable en ordre croissant : proposez un algorithme plus efficace que le précédent et évaluez sa complexité.

Question 2 : Structure de données (14 points)

L'objectif de cet exercice est de **concevoir** et de **réaliser** une structure de données permettant :

- d'insérer un nouvel élément ;
- de consulter et/ou d'extraire l'élément le plus ancien présent dans la structure ;
- de consulter et/ou d'extraire l'élément le plus récent présent dans la structure.

Par convention, on appellera cette structure *Chenille*.

1. Opérations (2 pts)

On identifie 7 opérations nécessaires à la manipulation d'une structure de *Chenille* : *Vide*, *Insérer*, *EstVide*, *Tête*, *Queue*, *Étêter*, *Équeuter*.

Précisez le profil de chaque opération (types des arguments, type de retour). Comme vu en cours, on privilégie un point de vue *fonctionnel* (pas d'effet de bord).

2. Préconditions (1 pt)

Quelles opérations ne peuvent pas être utilisées lorsque la *Chenille* est vide ?

3. Axiomes (4 pts)

Recopiez et complétez les 8 égalités suivantes, qui permettent de spécifier le comportement des opérations choisies.

$\text{EstVide}(\dots) = \dots$

$\text{EstVide}(\dots) = \dots$

$\text{Tete}(\text{Insérer}(e, c)) = \dots$

$\text{Tete}(\dots) = \dots$

$\text{Queue}(\dots) = \dots$

$\text{Étêter}(\text{Insérer}(e, c)) = \dots$

$\text{Étêter}(\dots) = \dots$

$\text{Équeuter}(\dots) = \dots$

4. Simulations (2 pts)

Est-il possible de simuler une pile à l'aide de cette structure *Chenille*, et si oui comment ?

Est-il possible de simuler une file à l'aide de cette structure *Chenille*, et si oui comment ?

5. Implémentation (5 pts)

Écrivez une implémentation du type *Chenille* basée sur un tableau d'éléments, et si besoin un nombre fini de variables.

Pour cette question, vous utiliserez des effets de bord.

Un schéma préliminaire vous sera d'un grand secours et sera apprécié par le correcteur.

Plus vous parviendrez à réaliser d'opérations en temps constant, plus votre solution sera valorisée.

Bonus : Comme notre tableau est de taille limitée, la *Chenille* pourra être « pleine » ; décrivez le nombre maximal d'éléments qu'on peut mémoriser (en fonction de la taille du tableau), et prévoyez une condition précise pour afficher un message d'erreur si on cherche à insérer un élément de trop.