

IFT2015-H26 - Quiz #2 - 22 janvier 2026

Nom:

Prénom:

Matricule:

Directives:

- Répondez directement sur cette feuille
- Aucune documentation ni appareil électronique permis.

(5 points) Question 1 : Considérez une liste de favoris implémentée avec une liste positionnelle, où chaque élément stocke un compteur représentant son nombre de consultations.

Donnez la complexité associée à chaque énoncé suivant :

Accéder à la première position d'une liste positionnelle

$O(1)$

Avancer à la position suivante

$O(1)$

Insérer à une position

$O(1)$

Supprimer à une position

$O(1)$

Chercher un élément

$O(n)$

Accéder au compte d'un élément à une position

$O(1)$

Mise à jour après une consultation avec la politique garder-triée

$O(n)$

Obtenir le top-k avec la politique - garder-triée

$O(k)$

Mise à jour après une consultation avec la politique "move-to-front"

$O(1)$

Obtenir le top-k avec la politique "move-to-front"

$O(kn)$

(5 points) Question 2 : Indiquez dans le rectangle ci-dessous quelle(s) affirmation(s) parmi les suivantes sont correctes.

- A. Une pile implémentée par un tableau dynamique a `push` et `pop` en $O(1)$ amorti.
- B. Une file implémentée par une liste simplement chaînée (sans pointeur sur le dernier noeud) a `enqueue` et `dequeue` en $O(1)$.
- C. Une deque implémentée par une liste doublement chaînée permet toutes les opérations en $O(1)$.
- D. Une deque implémentée par un tableau circulaire permet toutes les opérations en $O(1)$ amorti.
- E. Une pile implémentée par une liste simplement chaînée (avec pointeur sur le dernier noeud) nécessite un parcours de la liste pour `pop`.
- F. Dans une liste simplement chaînée circulaire, on peut parcourir tous les éléments à partir de n'importe quel noeud, sans jamais rencontrer `null`.
- G. Une file implémentée avec une liste circulaire peut réaliser `enqueue` et `dequeue` en $O(1)$ avec un seul pointeur vers le dernier élément.
- H. La classe `java.util.Stack` est recommandée pour les nouvelles implémentations de piles en Java.
- I. Dans une deque implémentée avec un tableau circulaire, l'arithmétique modulo permet d'éviter les décalages de tous les éléments lors des insertions et suppressions.
- J. Dans un tableau circulaire de taille N , l'expression $(\text{index} - 1) \% N$ donne toujours un indice valide du tableau.

A, C, D, G, I

(5 points) Question 3 : Indiquez dans le rectangle ci-dessous quelle(s) affirmation(s) parmi les suivantes sont correctes.

A. On utilise `LinkedList` comme file : le programme peut violer les invariants de la file et produire des bogues non reproductibles.

B. On utilise une file thread-safe (eg. `ConcurrentLinkedQueue`) avec le code

```
if( !q.isEmpty() ) q.remove();
```

La structure est thread-safe mais le programme peut rester logiquement incorrect.

C. On utilise une file protégée par un `synchronized` ou par un verrou (`lock`): cela garantit que tous les éléments produits seront consommés.

D. On utilise `BlockingQueue` avec une terminaison par pilules empoisonnées : la coordination entre producteurs et consommateurs est intégrée à la structure.

E. Une pilule empoisonnée est un mécanisme de synchronisation équivalent à `wait()/notifyAll()`.

A, B, D