

## IFT 2015 E21

### Devoir 1.

10/10, soit 10% de la note finale.

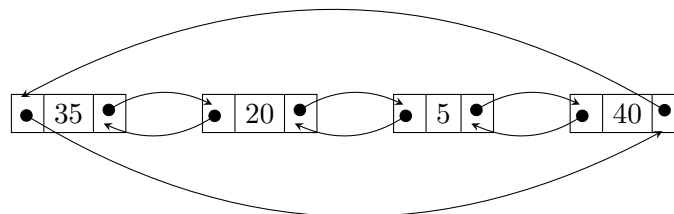
Les 10 points “Partie pratique” pour le cours E21 seront presque certainement distribués de la façon suivante : Devoir #1 (1 point), Devoir #2 (3 points), Devoir #3 (6 points).

Les réponses doivent être remises dans un document format *pdf*. Il y aura un bonus de 1 point pour un document préparé avec  $\text{\LaTeX}$  (Devoir 1 seulement. La note maximale pour le Devoir 1 est donc 11 sur 10.)

## 1 Partie Pratique (1 point)

Une liste doublement chaînée est dite *circulaire* lorsque, plutôt que d’être nul, le pointeur *next* du dernier noeud contient l’adresse du premier noeud, et le pointeur *prev* du premier noeud contient l’adresse du dernier noeud. Ainsi, un parcours dans la liste n’aboutira jamais à un pointeur nul : lorsque nous arrivons à ce qui devrait être la fin, le prochain pointeur nous ramène au point de départ et nous pouvons continuer le parcours de façon cyclique.

Voici de quoi aurait l’air une telle liste contenant [35, 20, 5, 40] et où le noeud contenant 35 est le noeud de départ :



Pour ce premier devoir, vous devez implanter cette structure de données en *Java*. Un fichier squelette nommé `ListeCirculaire.java` vous est déjà fourni. Des instructions plus détaillées à propos des méthodes à implanter sont décrites dans les commentaires du fichier. Veuillez remettre votre version complétée du fichier sur StudiUM, avec les noms et matricules des auteurs en commentaire au début du fichier.

Ne changez pas le nom du fichier, ni les signatures des méthodes déjà présentes. Si vous décidez d’ajouter une ligne “package” au début du fichier, retirez-la dans la version que vous soumettez. Assurez-vous que votre fichier compile : il est difficile de donner des points pour du code qui ne compile pas !

Un fichier `Main.java` contenant quelques tests vous est aussi fourni. Vous pouvez l’utiliser comme bon vous semble pour tester votre implémentation. Vous n’avez pas à le remettre, et nous ne tiendrons pas compte des modifications que vous y ferez : nous ne regarderons que votre fichier `ListeCirculaire.java`.

## 2 Partie Théorique (9 points)

1. (1 point)

Je vous ai donné des définitions formelles de  $T_{best}$ ,  $T_{avg}$  et  $T_{worst}$ .

Démontrez formellement que  $T_{best}(N) \leq T_{avg}(N) \leq T_{worst}(N)$ .

(Les preuves sont faciles et courtes.)

2. (1 point)

Si vous êtes plus à l'aise avec cela, vous pouvez remplacer le symbole epsilon par un signe d'égalité.

(a) Écrivons  $f < g$  pour noter que  $f(N) \in O(g(N))$  mais  $g(N) \notin O(f(N))$ .

Trouvez trois fonctions  $f$ ,  $g$  et  $h$  telles que

$$f < g < h$$

et trois fonctions  $T_{best}$ ,  $T_{avg}$  et  $T_{worst}$  telles que

$$T_{best}(N) \leq T_{avg}(N) \leq T_{worst}(N), \quad N \geq 0$$

et

$$T_{worst} \in ?(f(N)), T_{avg} \in ?(g(N)), T_{best} \in ?(h(N)).$$

Vous avez le droit de remplacer chaque point d'interrogation par le symbole  $O$  ou  $\Omega$ , mais vous ne pouvez pas faire le même choix trois fois.

(b) Trouvez trois fonctions  $f$ ,  $g$  et  $h$ , et une combinaison quelconque de  $O$  et  $\Omega$  pour remplacer les trois points d'interrogation pour définir les trois classes  $?(f(N))$ ,  $?(g(N))$  et  $?(h(N))$ , mais tel qu'il n'est pas possible de trouver  $T_{best}$ ,  $T_{avg}$  et  $T_{worst}$  telles que

$$T_{best} \in ?(f(N)), T_{avg} \in ?(g(N)), T_{worst} \in ?(h(N)) \text{ et}$$

$$T_{best}(N) \leq T_{avg}(N) \leq T_{worst}(N), \quad N \geq 0.$$

L'exemple d'impossibilité doit impliquer les deux fonctions  $T_{avg}$  et  $T_{worst}$ , et il faut donner la preuve de l'impossibilité.

3. (2 points)

(a) Weiss, Exercice 2.2, page 50, partie (c) seulement. (Ajoutez le mot “necessarily” avant le mot “true”.) Donnez une preuve ou donnez un contre-exemple simple.

(b) Weiss, Exercice 2.11, page 52.

4. (1 point)

Weiss, Exercice 2.7, page 50, partie (a) seulement, et seulement les cas (2), (3), (4) et (5).

5. (1 point)

Weiss, Exercice 2.14, p. 52.

6. (2 points)

Weiss, Exercice 4.6, page 161.

7. (1 point)

Weiss, Exercice 4.9, page 161.

(Faire les deux cas : échange avec prédécesseur, échange avec successeur.)

À réaliser en équipes de 1 ou 2. À remettre le 2 juin, 2021, avant 9:00. Les solutions seront affichées le 2 juin. Les devoirs en retard ne seront pas acceptés.