

# Devoir 3

IFT2015

Hiver 2025

## Consignes pour le Devoir

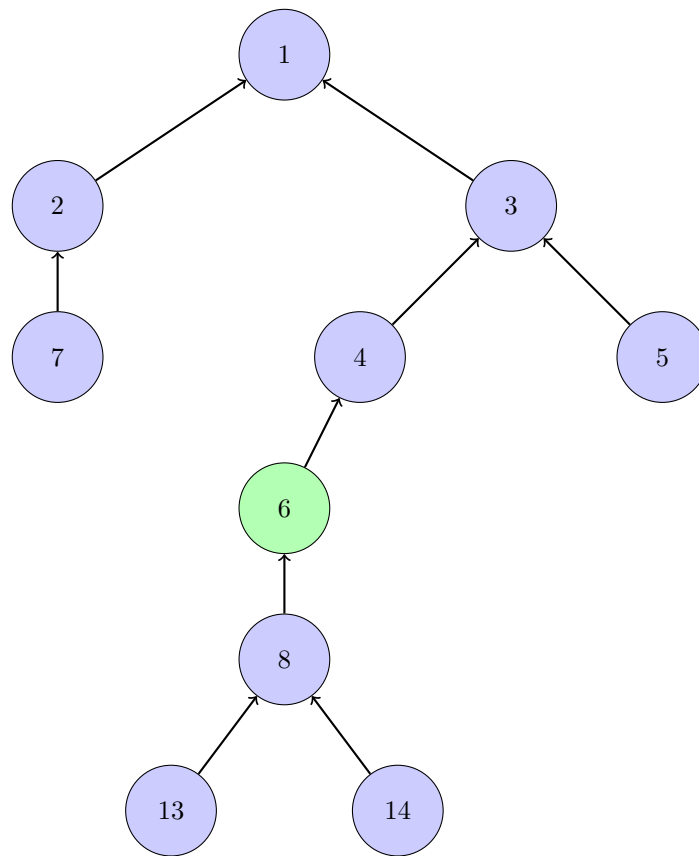
Implémentez votre solution dans un fichier nommé `reponse_DA_DA.java`, où vous remplacerez `DA` par vos numéro de `DA`. Le devoir est à faire en équipe de 2, une seule personne fait la remise pour l'équipe. Tout votre code doit être dans ce fichier : vous pouvez créer de nouvelles classes ou fonctions, mais ne modifiez pas les signatures des fonctions `question_1`, `question_2`, etc., qui seront utilisées pour les tests. Utilisez le fichier `main` fourni pour tester vos réponses, mais notez que des tests supplémentaires seront effectués. Commentez clairement votre code, utilisez des noms de variables explicites, et n'ajoutez aucune bibliothèque externe. Vous devez absolument créer la structure de données sans utiliser une bibliothèque pour répondre à la question ; des points seront accordés pour l'implémentation de la structure et non juste pour la réponse. Assurez-vous que votre fichier compile avant de le soumettre, car un fichier non fonctionnel sera fortement pénalisé. Remettez uniquement `reponse_DA_DA.java`. Assurez-vous également que cela fonctionne avec le fichier `main` qui n'a pas été modifié, car les changements dans le fichier `main` ne seront pas corrigés. Pour finir, vos solutions doivent s'exécuter dans un temps raisonnable (moins d'une minutes).

## Q1: Contamination des égouts (4 points)

Un incident est survenu dans les égouts de la ville : un produit chimique dangereux a été accidentellement déversé dans une canalisation. Ce produit se propage en aval, contaminant toutes les zones connectées en descendant dans le réseau. Votre mission est de déterminer quelles canalisations sont contaminées en aval à partir du point d'origine. Les égouts de la ville forment un arbre où chaque canalisation est connectée à une ou plusieurs autres. Une canalisation principale collecte les eaux provenant d'autres canalisations connectées en amont. Les données d'entrée décrivent la structure des égouts et la position initiale du produit chimique. La première ligne est un entier indiquant le numéro de la canalisation où le produit chimique a été déversé. La dernière ligne contient uniquement l'entier  $-1$ , qui marque la fin des données. Toutes les autres lignes contiennent une liste d'entiers. Le premier entier représente la canalisation en aval, et les autres entiers correspondent aux canalisations en amont qui se jettent dans cette première canalisation. Les entrées ne sont pas nécessairement ordonnées. Par exemple, les deux entrées suivantes :

6	6
1 2 3	1 2 3
3 4 5	4 6
4 6	2 7
2 7	3 4 5
6 8	6 8
8 13 14	8 13 14
-1	-1

forment toutes deux le même réseau d'égouts suivant :



Pour ce réseau, si le produit chimique est déversé dans la canalisation 6, la sortie attendue serait :

$$6 \rightarrow 4 \rightarrow 3 \rightarrow 1$$

Votre programme doit afficher la liste des canalisations contaminées en aval, dans l'ordre où elles sont atteintes par le produit chimique. Pour ce faire, vous devez absolument construire un arbre représentant le réseau.

## Q2: La bijouterie (2 points)

Pour payer vos études à l'Université de Montréal, vous décidez de faire un vol dans une bijouterie (À ne pas faire). Vous êtes dans la bijouterie, l'alarme a sonné et vous avez peu de temps avant que la sécurité n'arrive. Votre objectif est clair : prendre les objets les plus précieux et partir rapidement. Dans la panique, vous décidez de sortir votre ordinateur pour créer un programme qui vous permettra de sélectionner les bijoux les plus précieux plus rapidement. Vous vous souvenez qu'un tas (heap) a été conçu précisément pour trier des éléments rapidement et efficacement. "Parfait, ça va m'aider à optimiser mon vol tout en révisant mes connaissances sur les structures de données." Vous vous mettez alors à coder.

**Consignes :** Implémentez un monceau avec des fonctions pour insérer et extraire des éléments. Trier une liste en utilisant le monceau. Insérez les éléments un par un dans le monceau, puis extrayez-les dans l'ordre trié.

### Exemple d'Entrée

```
120$ Bracelet
200$ Bague
250$ Collier
70$ Boucles d'oreilles
100$ Broche
50$ Montre
300$ Pendentif
350$ Diadème
```

### Exemple de Sortie

```
Diadème
Pendentif
Collier
Bague
Bracelet
Broche
Boucles d'oreilles
Montre
```

Implémentez le monceau sans utiliser de bibliothèques externes et vérifiez que votre solution fonctionne bien.

### Q3: Les pommes à Newton (4 points)

Vous vous retrouvez en prison avec Newton après votre tentative de vol de bijouterie. Vous êtes surpris que cela ait échoué, malgré l'algorithme rapide que vous avez implémenté, qui était pourtant en  $O(n \log(n))$ . Vous demandez à Newton pourquoi il est ici, et il vous répond qu'il a vidé des produits chimiques dans les égouts d'une ville pour une expérience, ce qui a conduit à son arrestation par la police. Il vous demande également votre aide. Newton adore trier ses pommes et a besoin de votre aide pour créer un arbre binaire de recherche équilibré (AVL) à partir d'une liste d'opérations. Il vous fournira une liste d'opérations, chacune étant soit un ajout ("A") soit une suppression ("S"). Par exemple, l'opération "A 3 #c20606" ajoute une pomme numérotée 3 avec la couleur #c20606 à l'arbre, tandis que "S 3" supprime la pomme numérotée 3. Après chaque opération, vous devez maintenir l'arbre équilibré et, pour les suppressions, retourner le nombre de couches dans l'arbre, qui sera 0 si l'arbre est vide. Par exemple, après avoir supprimé la pomme numérotée 3, le résultat est "#c20606 0". Il est crucial d'implémenter la structure de données demandé et de vérifier qu'elle fonctionne correctement avec le fichier de test fourni.