

# IFT2015 Hiver 2009 — Devoir 2

Miklós Csűrös

4 février 2009

À remettre au cours le 11 février. Remettez un rapport écrit. Un rapport soumis par courriel (à csuros@iro...) en format PDF vaut  $\pi$  points de boni.

## 1 Suppression (25 points)

Considérez la squelette suivante d'une classe utilisant une liste chaînée.

```
public class Liste_du_devoir2
{
    static class Element // pour stocker les objets sur la liste
    {
        private Object valeur ;
        private Element prochain;
        Element(){valeur=null; prochain=null;}
    }

    private Element tete;
    public Liste_du_devoir(){ tete = null;} // liste vide

    public void supprimer(Object O)
    {
        // a ecrire ...
    }
}
```

La liste est une liste simplement chaînée, sans sentinelle et la fin de la liste est dénotée par `null`. Dans cet exercice, vous devez écrire le code pour la méthode `supprimer` qui cherche le premier élément sur la liste dont la valeur est égal à `O` et supprime cet élément de la liste. Ce qui se passe lorsque l'objet `O` n'est pas sur la liste est à vous à décider : expliquez votre décision. Analysez le temps de calcul de l'opération (dans le pire cas, sur une liste de taille  $n$ ).

## 2 Calcul à pile (10 points)

Faites la conversion de l'expression suivante vers la notation post-fixe :

$$(((1 * ((2 + 3) * 4)) + (5 * 6)) - (7 + 8)).$$

Quel est le contenu de la pile en Postscript après l'exécution du code suivant ?

```
1 2 3 4 5 6 add mul 7 add mul mul add 8 sub
```

## 3 Minipile (10 points)

Considérez une pile avec les opérations usuelle `pop` et `push`. Montrez comment on peut aussi implanter l'opération `getMin` qui retourne l'élément de valeur minimale sur la pile (supposons que c'est une pile sur des nombres). Votre solution doit supporter toutes les trois opérations en temps  $O(1)$ .

## 4 Deux c'est mieux (15 points)



Expliquez comment implanter une queue en utilisant deux piles. Montrez la [pseudo]code pour l'implantation des opérations `enqueue`, `dequeue`, `isEmpty`. Analysez le temps de calcul de chacune des opérations.