Rapport pour le Projet 3 d'Algorithmique

OOMS Aurélien BA1 INFO

9 mai 2011

Table des matières

1 Introduction

Dans le cadre du cours d'algorithmique, il nous a été demandé de créer une structure, un « Counting Tree », permettant de stocker et d'afficher les différents bulletins de vote d'une élection et le résultat du scrutin de cette même élection.

Les sections qui suivent décrivent le fonctionnement des différentes classes et fonctions mises en place.

2 Class Urn

Une urne est une structure initialement vide, capable d'accueillir un nombre illimité de bulletins de votes offrant chacun la possibilité de voter pour les candidats.

2.1 __init__(self)

On initialise l'urne avec un attribut __ballot qui est une liste pouvant contenir des bulletins ayant le même attribut numberOfPossibleChoices [1].

2.2 addBallot(self)

On utilise la méthode append des listes python pour ajouter un bulletin.

2.3 tally(self)

Renvoie le résultat du scrutin, c'est-à-dire l'urne de départ compressée au maximum [2].

2.4 assertInvariant(self)

Vérifie que tous les bulletins de l'urne ont le même attribut numberOfPossibleChoices.

2.5 getCompressed(self,f)

Renvoie une nouvelle urne, image par compression de celle passée en paramètre. Cette méthode prend un paramètre f, facteur de compression. Si f est égal à 2, l'urne résultat sera compossée de bulletins chacun égaux à la somme d'au maximum 2 bulletins [3] de l'urne de départ.

2.6 getBallot(self,i)

Cette méthode prend un paramètre i, et renvoie le bulletin d'indice i. Permet le parcours linéaire de la liste, utile dans d'autres méthodes.

1. Nombre de candidats.
2. Le facteur de compression est égal au nombre de bulletins.
3. Chacun des bulletins est pris une et une seule fois.

3 Class CountingTree

La classe ne correspond pas exactement à celle demandée dans l'énoncé. Pour les justifications, voir commentaires code.

3.1 toRegularTree(self,numberOfChildren = 2)

Méthode qui renvoie un arbre binaire illustrant le résultat du scrutin.

3.2 __init__(self,urn,m)

Initialise l'arbre en lui donnant les attributs __urn, l'urne du scrutin, et m, le nombre de fils des noeuds de l'arbre comptage.

3.3 tally(self)

Renvoie le résultat de la méthode tally sur __urn.

3.4 __repr__(self)

Crée l'arbre comptage à partir de la classe Forest et l'affiche à l'aide de la méthode d'affichage de la classe Forest.

3.5 Class Queue

Classe vue au cours et utilisée dans le cadre de ce projet.

3.6 Class Forest

Classe vue au cours et utilisée dans le cadre de ce projet.

4 Conclusion

Jusqu'à présent, les projets réalisés étaient le fruit d'un travail individuel. Ce projet-ci demande de comprendre et d'utiliser des structures déjà implémentées par d'autres et limite donc les possibilités pour résoudre le problème. Ce n'en est pas moins un exercice intéressant.

5 Annexe

```
acolyte@ubuntu-VirtualBox:~/Documents/Projet Algo 3$ python CountingTree.py
**********************************************
Level 0
[26]
**********************************************
Level 1
[24] [2]
**********************************************
Level 2
[10] [14] [2]
**********************************************
Level 3
[5] [5] [5] [9] [2]
**********************************************
Level 4
[3] [2] [2] [3] [3] [2] [5] [4] [2]
**********************************************
Level 5
[2] [1] [1] [1] [1] [1] [1] [2] [2] [1] [1] [1] [3] [2] [3] [1] [2]
**********************************************
```

FIGURE 1 – Capture d'écran du terminal.

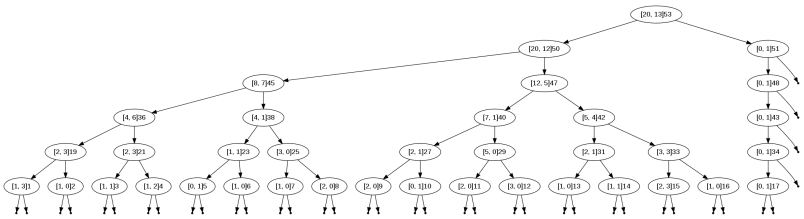

FIGURE 2 – Image résultat de la commande «python Draw.py».