

SAE 1.02

Programmes de vote

Groupe : HSSINI Ayoub, MANKAI Adam, MALLET Mathias
et JAMBERT Aurèle

Introduction

Dans le cadre de la SAE 1.02, il nous a été demandé de réaliser des systèmes de votes en c++. Nous avons donc réalisé des programmes qui simulent les systèmes de vote suivants :

Vote majoritaire à 2 tours

Méthode d'Hondt de la moyenne la plus forte

Vote préférentiel (borda amélioré)

Le vote majoritaire à 2 tours est un vote où on demande aux votants de choisir un des candidats, à la fin de ce premier tour nous avons 2 candidats majoritairement votés, il y a donc un deuxième tour qui renvoie le vainqueur parmi les 2 candidats. Par exemple, en France pour élire le président , nous réalisons un vote majoritaire à 2 tours.

La méthode d'Hondt est un mode de scrutin qui sert à répartir les sièges dans une assemblée de manière proportionnelle aux votes reçus par chaque parti. Pour faire simple, on divise le nombre de voix de chaque parti successivement par 1, 2, 3, 4 etc. (ce sont les diviseurs d'Hondt.). Les quotients obtenus (les moyennes) sont classés par ordre décroissant, et les sièges sont attribués aux partis qui ont les plus grandes moyennes. Par exemple, lors de l'élection des eurodéputés, on utilise la méthode d'Hondt pour attribuer les sièges aux députés de chaque pays de l'Union européenne.

Le vote préférentiel (ou vote par classement) permet aux électeurs de classer les candidats par ordre de préférence au lieu de n'en choisir qu'un seul. Le candidat ayant le classement moyen, le plus faible ou le score le plus élevé (selon la variante utilisée) est généralement déclaré vainqueur. Nous avons utilisé comme méthode un bord amélioré, si nous prenons les 4 parfums de glace par exemple, selon les votes, celui qui est premier aura 4^2 points, le deuxième 3^2 points, le troisième 2^2 points et le quatrième 2^1 points.

Nous additionnons donc tous les points de tous ceux qui ont voté selon leur classement.

Jeux de test et Validation

Nous avons testé nos programmes avec les entrées et oracles donnés par M.Casali. Cependant, pour la méthode d'Hondt, nous avons dû changer la syntaxe des 2 entrées proposées pour les adapter à nos codes. Les oracles ont notamment été changés pour la méthode d'Hondt pour renvoyer les résultats, soit le nombre de "sièges" pour chaque candidat.

De même pour le vote majoritaire à 2 tours, nous les avons changés pour que les votants puissent voter 2 fois.

Difficultés

La lecture et l'exploitation des flux ont été un obstacle majeur, plus de 2 heures de travail ont été nécessaires à la simple conception des programmes de traitement des données (voix et candidats).

Nous sommes finalement parvenus à comprendre comment faire passer les entrées avec les commandes suivantes :

```
./"Nomdufichier.cpp" < entree1.txt
```

Concernant le reste, la logique des programmes a été compliquée à mettre en place, mais nous a néanmoins posé moins de problèmes que les flux...

L'élément le plus défectueux n'a pas été un programme, mais bien un concept : l'organisation. En effet, nous avons sous-évalué le temps nécessaire à certains programmes et surtout, nous avons eu du mal à nous coordonner pour proposer un travail fluide.

Rôles de chacun

Ayoub et Aurèle ont réalisé le programme d'Hondt, ils ont fait les recherches et le code de la méthode. Ayoub s'est principalement occupé de la structure du code et Aurèle le codage, même s'ils ont codé les deux chacun de leur côté, ils ont fini par choisir le code d'Aurèle.

Le vote majoritaire à 2 tours et le vote préférentiel ont été réalisés par Mathias et Adam, avec des recherches et du temps de réflexion sur le code.

Le rapport et le diaporama ont été réalisés par Ayoub et Aurèle.