Projet Personnel 2020 Été

Application d'affectation de candidats après des concours multiples

Zihao HUA, 3ème année Département Informatique, INSA de Lyon

1. Contexte du problème

Dans le cas des affectations des étudiants en première année des études de santé (PACES), les candidats passeront plusieurs concours dans divers secteurs et obtiendront plusieurs résultats de classement différents. Alors que l'affectation de filière devient plus complexe car le rang de classement, autrement dit le critère d'attribution, n'est plus unique. Il faut donc trouver une méthode pour décider de l'affectation des étudiants dans les différentes filières en prenant en compte à la fois les classements des étudiants aux différents concours et leurs voeux entre les différentes filières.

2. Objectif du projet

À l'aide de l'algorithme Gale et Chapley (algorithme de mariage stable) , nous pouvons trouver une solution adéquate pour résoudre ce problème d'affectation multiple. L'objectif de ce projet est donc mettre en oeuvre cet algorithme en langage C++, ensuite développer une application web qui permet aux utilisateurs étudiants de :

- Gérer son compte personnel et mettre à jour sa liste des vœux avant la date de fermeture de la saisie des vœux.
- Consulter son résultat d'affectation après que le calcul se termine.

3. Descriptions et Spécifications

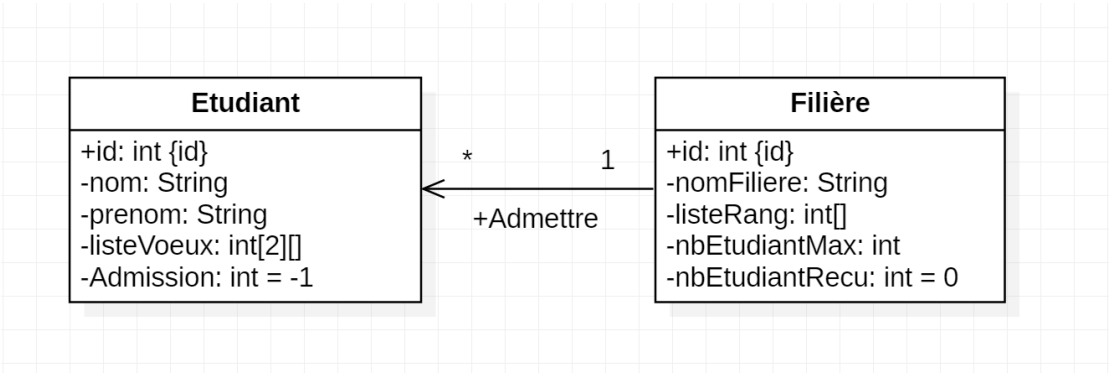
D'après le pseudo code sur le site wikipedia de l'algorithme Gale et Shapley :

```
Entrée : Deux ensembles finis M (d'hommes) et W (de femmes) de cardinal n ;
        Une famille L de relations de préférences ;
Sortie : Un ensemble S de couples engagés (homme ; femme) ;
fonction mariageStable {
    Initialiser tous les m ∈ M et w ∈ W à célibataire
    tant que ∃ homme célibataire m qui peut se proposer à une femme w {
        w = femme préférée de m parmi celles à qui il ne s'est pas déjà proposé
        si w est célibataire
            (m, w) forment un couple
        sinon un couple (m', w) existe
            si w préfère m à m'
                (m, w) forment un couple
```

```
                m' devient célibataire
            sinon
                (m', w) restent en couple
        }
        Retourner l'ensemble S des couples engagés
}
```

Tout d'abord, chaque rôle dans le scénario doit posséder une liste de préférence pour commencer l'affectation. Dans notre structure de données, il s'agit d'un tableau **int** mémorisant sa préférence (les identifiants des étudiants/filières) par ordre descendante. De plus, pour les « prétendants » (c'est-à-dire les étudiants dans notre cas) , leur tableau de voeux est à deux dimension afin d'enregistrer les refus.

Ensuite, les « couples » formés sont enregistrés dans un tableau d'admission **int** à deux dimension de taille nbEtudiantTotal*2, il sera parcouru à la fin pour que le résultat soit enregistré dans une base de données sous une forme plus complète. Je le trouve plus efficace que l'on enregistre temporairement les couples dans les classes Etudiant qui sont pénibles à tout parcourir.



Quant à la conception des classes, il nous faut encore ajouter 2 attributs : nombre d'étudiants maximum et nombre d'étudiants reçus qui correspond à notre cas.

Dans la base de données, on y trouvera 3 tableaux : **Etudiant, Filière et Admission**, ce dernier est donc le tableau d'association entre les étudiants et les filières, il sera complété après que l'algorithme se termine. Le mot de passe des comptes étudiants se trouve dans le tableau **Etudiant.**

Les méthodes de classes et les détails de a partie Front-End seront précisés dans les rapports à suivre.

4. Plan du développement

Je compte diviser le développement en 3 périodes :

- 15/07/20 ~ 29/07/20 : Mise en oeuvre des codes C++, dont les premiers 2-3 jours consistent à implémenter les codes du problème original (mariage N à N stable). Une fois que cela s'est fait, je commencerai à les modifier ou reconstruire pour qu'elles s'adaptent à l'objectif.
- 29/07/20 ~ 12/08/20 : Développement de la partie Front-End.
- 12/08/20 ~ 26/08/20 : Tests, débogages et rédaction du rapport.