

LaTeX report of Seat-Allocation

Aman, Anmol, Pranjal

October 31, 2014

Table of Contents

1. About the Problem
 - GaleShapley Algorithm
 - MeritOrder Algorithm
2. Experience with the Course
3. Conclusion

Why a Project ?

Software Systems lab is a course which is meant to explore all sorts of softwares

The idea is to cover a wide range of softwares which will surely help us to command respect over non - CS majors

But that is a secondary reason

The primary reason is to experience computer science - to experience learning - to experience exploring!

The project is meant to teach us "team - work" and collaboration

What's there in it ?

This project aims at making a Program for seat-allocation in IIT-JEE using java.

We intend to create a Program which when provided with preferences of candidates, ranklist and qouta for each course, gives back the branch he received

The idea is to use the two algorithms provided by sir.

It will contain several arraylist and hashmap for fast access of data to make the program effecient.

Proper data structure and data hiding(private data with public functions to access) is used as part of object oriented implementation programing.

About the Problem

The goal of this lab assignment is to implement two algorithms for college admissions. Each algorithm takes into account the preferences of students and the merit lists of the colleges in order to best allocate seats.

There are many candidates who desire admission and many programmes to be allotted.

(A programme is a particular stream in a particular institute, e.g., CSE in IITB, EE in NIT Nagpur etc.)

A programme has a quota for each category. For example, CSE in IITB may accept 40 students in General Category;

15 in OBC, 5 in SC, 5 in ST etc.

The goal is to allot seats to students. It is desirable for a student to get the highest preference possible; while being fair to the other students based on the merit list. That is,

1. Fairness: Suppose candidate x is allotted program p . Then for any other candidate y such that y has a better (smaller) rank than candidate x in the merit list of p , the allocation of y should be p or some other program that y prefers to p .
2. Optimality: For any candidate x , there does not exist any other allocation that satisfies the Fairness property, and provides x with an allocation she prefers to (x) .

Gale-Shapley Algorithm

Initially DS allocation takes place:

- Firstly all the candidates present in the DS Category apply for their first preference.
- Now we go through all the institutes maintaining a common rejectionlist and take the top two or more(depends on same rank) according to their GE rank and add to waitlist.
- Those who were not waitlisted are added to the rejection list.
- Now we go through the rejection list and for all the candidates present we call the nextDSVirtualProgramme() which increases the pointer to the next program in preference list.(Pointer points to the program to be applied)
- Now we check if the rejectionlist is empty or not. If it is not empty we repeat the process and everyone applies to their current program(to which the pointer points) except if they have exhausted their preference list, then those candidates dont apply.
- If the rejection list is empty, we get out of the loop and those who were not waitlisted now act as their category(e.g. GE, SC, ST-PD, etc) and all their pointers are reset.

Now normal allocation takes place:

- Firstly all the candidates except selected DS Category and F category apply for their first preference.
- Now we go through all the VirtualProgramme(for each program we go through all the 8 category VirtualProgramme) maintaining a common rejectionlist and take the top quota no. of candidates or more(depends on same rank) according to their rank in the meritlist of the VirtualProgramme they applied in rank and to waitlist.
- Those who were not waitlisted are added to the rejection list.
- Now we go through the rejection list and for all the candidates present we call the nextVirtualProgramme() which increases the pointer to the next program in preference list.(Pointer points to the program to be applied)
- Now we check if the rejectionlist is empty or not. If it is not empty we repeat the process and everyone applies to their current program(to which the pointer points) except if they have exhausted their preference list, then those candidates dont apply.
- If the rejection list is empty, we get out of the loop.

Now F allocation takes place:

- Firstly all the candidates present in the F Category apply for their first preference.
- Now we go through all the VirtualProgramme(for each program we go through all the 8 category VirtualProgramme) maintaining a common rejectionlist.
- If there is an empty seat then the candidate gets it or else we compare the rank of the last candidate with our candidate and if he is better or equal he gets a seat(resulting in extension of quota) else he is added to rejectionlist.
- Now we go through the rejection list and for all the candidates present we call the nextFVirtualProgramme() which increases the pointer to the next program in preference list.(Pointer points to the program to be applied)
- Now we check if the rejectionlist is empty or not. If it is not empty we repeat the process and everyone applies to their current program(to which the pointer points) except if they have exhausted their preference list, then those candidates dont apply.
- If the rejection list is empty, we get out of the loop and get the desired result.

Merit-Order Algorithm

Initially DS allocation takes place:

- We go through DS ranklist(based on GE rank only) and take candidates one by one.
- The candidate keeps on applying in his preference list untill he gets selected or he exhausts his list.
- For each program we check the institute seats left, if there is a seat he gets it (or there is same rank so quota extension can take place) and we call the next candidate else he tries the next program in his list (and checks that programs intitute seats).
- Once we go through the whole list DS allocation is complete and we reset the pointer of all those who were not selected.

Now Phase-I allocation takes place:

- We go through commonRankList and take candidates one by one.(only those who are not already waitListed and not any F Candidates)
- The candidate keeps on applying in his preference list untill he gets selected or he exhausts his list.
- For each program if there is a seat he gets it (or there is same rank so quota extension can take place) and we call the next candidate else he tries the next program in his list.
- Once we go through the whole list Phase-I is complete.

Now de-reservation takes place:

- We go through all the programs and if there is any OBC, GE-PD or OBC-PD seats they are allotted to GE. If any SC-PD/ST-PD seats are left they are allotted to SC/ST.

Now Phase-II allocation takes place:

- First we reset the pointer of non-selected candidates.
- It is similar to Phase-I i.e. we go through commonRankList and take candidates one by one.(only those who are not already waitListed and not any F Candidates)
- We follow rest steps of Phase-I

Now F allocation takes place:

- We go through F rankList and take candidates one by one.
- The candidate keeps on applying in his preference list untill he gets selected or he exhausts his list.
- For each program if there is a seat he gets it or if there is a candidate with worse or equal rank to him then he gets the seat.
- Once we go through the whole list our algo is complete and we get the desired result.

JEE Choice filling Web Application

DJANGO - A Completely new Experience!

- We designed a JEE choice filling Web application which lets a student fill his/her choices for JEE counselling
- We used Python for the coding part - python in Django framework
- In this project report, we have tried to elaborate our project work
- We have also mentioned the references that helped us during the course of the project work

Features of the web application :

The web application is meant to provide the following functionality to the user :

- A user can log into his/her account in the web application
- A user can fill his/her choices in the web application - the application takes care of the fact that a particular choice is filled only once
- A user can edit his/her existing choices
- A user can delete his/her existing choices
- A user can check the last year's opening as well as closing ranks for a particular programme
- A user can also select an institute and category, and the web application will help the user to decide which branches he/she is likely to get

Platforms used in this project :

- We used the Django framework
- We did all the coding part in python
- The web app itself is in HTML
- We also employed CSS

Things learnt in this project :

During the course of this project, we learnt a lot. We learnt the following :

- How to learn and understand a completely new software - django, only and only with the help of online tutorials
- How to work as a team in the development of a not so trivial project
- How to use internet efficiently to search for errors, syntaxes and common bug fixes
- How to manage time

1 Experience with the Project

Project Credits :

The following people were involved in the development of this project :

- 1.) Anmol Arora (Roll number 130050027)
- 2.) Pranjal Khare (Roll number 130050028)
- 3.) Aman Goel (Roll number 130050041)

Project Time Line :

The problem statement of the project was released by the start of October 2014

We started to work on it soon after it was released

The work was at it's peak during the Diwali Holidays (October 17th to October 26th)

We successfully completed the project well before time

Experience with the project :

Our schedule is usually very hectic and additionally we got a project work

So we were indeed tensed

But the team contains very efficient planners who know how to manage things well

So we used to plan and sit together for our project work

Regular planning and skilled management ensured a steady progress in the project

We did face a number of difficulties

For example, taking care of all the corner cases in java was really confusing

Also we were new to java so we used to use Google a lot for any sort of help

Django was a new platform to work with. So we were bound to get stuck

But, in the end, we did get solutions to our problems

Conclusion :

We did enjoy a lot, and learning Java and Django was a great experience in itself
We had chosen to drop the idea of Box2D and the 3 of us decided to rather do the
(Lab 10 + Lab 11) Pro Version project
We realised that Learning Python and Java would be of much more utility
It will be better to learn 2 highly popular programming languages
than just learning Box2D
Also, both of these languages are currently widely in use
So we decided that it was a better idea to learn python and java
Another important reason was the beauty of the problem statement of Lab 10 and 11
The problem statement wanted us to create a real time working web application
which was indeed fascinating
We had just learnt the Gale - Shapley algorithm in our Discrete Structures
course and a project which required an implementation of a recently learnt
algorithm sounded pretty cool to us
Java is a highly user friendly language to work with
It is fast, and is also highly efficient with a number of standard libraries that ease our task
Similarly, python is very easy to work with
Django framework is a bit difficult to understand but once understood well,
It can be used to make a wide variety of powerful web applications
All in all, we learnt a lot and we look forward to many other such projects in future