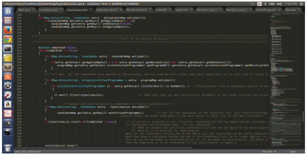
JEE Choice filling Web Application and Seat-Allocation: A CS251 Report by Group 11.

Anmol Arora
Roll No. 130050027 anmolarora59@gmail.com
Pranjal Khare
Roll No. 130050028 pranjalkhare7@gmail.com
Aman Goel
Roll No. 130050041 amangoelvsec@gmail.com

Oct 31, 2014

Overview

1. Introduction
2. Body
    - Part 1: Gale-Shapley Algorithm
    - Part 2: Merit-Order Algorithm
    - Part 3: Choice Filling Web Application
3. Conclusion
4. References



```
        }
        for(Map.Entry<String , Candidate> entry : dsCandidateMap.entrySet()){
            if(candidateMap.get(entry.getKey()).getAppliedUpto()==-1){
                candidateMap.get(entry.getKey()).setDSStatus(false);
                candidateMap.get(entry.getKey()).setAppliedUpto(0);
            }
        }
    ////////////////////////////////////////////////////////////////////////////////////////////
        /***************************************Start of the General Allocation***************************************/
    ////////////////////////////////////////////////////////////////////////////////////////////

        boolean completed=false;
        while(completed == false)
        {
            for (Map.Entry<String , Candidate> entry : candidateMap.entrySet())
            {
                if(entry.getValue().getAppliedUpto()!=-1 && entry.getValue().getNationality() && !(entry.getValue().getDSStatus())){
                    programMap.get(entry.getValue().currentVirtualProgramme().getProgramID()).get(entry.getValue().currentVirtualProgramme().getMeritListInd
                }
            }
            /** Till Now , all the candidates have applied to the programs. now the programs will either keep their applicants in the wait list or reject th

            for (Map.Entry<String , ArrayList<VirtualProgramme> > entry : programMap.entrySet())
            {
                for (ListIterator<VirtualProgramme> it = entry.getValue().listIterator(); it.hasNext(); )   /**ListIterator with no arguements points to the
                 {

                  it.next().filter(rejectionList);                    /** Make sure that you add rejectionList parameter in the filter function for Virtua
                }
            }
                  /** @doubt Can java variables be redefined? as "entry" variable here */
            for(Map.Entry<String , Candidate> entry : rejectionList.entrySet())
            {
                candidateMap.get(entry.getKey()).nextVirtualProgramme();        /** For all the candidates in the rejection list, call "nextVirtualProgr
                                                        present choice and makes them apply to the next choice in their list in the next iteration*/
            }
            if(rejectionList.size()==0){completed = true;}       /** If no one is rejected in this iteration, rejectionList will be empty and the iteration c
                                                        /** Iterations are terminated when , for all candidates:
                                                            1). When he reaches the end of his preference list and can not apply to any more program
                                                            2). When he is in Waitlist for some Program.
                                                        When all the candidates satisfy one of the above two, the algorithm can be safely terminate
                                                        Note that when a candidate reaches the end of his Preference list, he no longer applies to a
                                                        Thus he can not be added to rejectionList at all.(To be added in rejectionList, one needs to
                                                        Thus the algorithm is to be terminated iff the rejectionList is empty.*/

            rejectionList.clear();
```

The Purpose of this Report

We created a web application that helps a candidate to fill his/her choices for JEE counselling.

The Purpose of this Report

We created a web application that helps a candidate to fill his/her choices for JEE counselling. We also created a program in java that takes into account the preferences of students and the merit lists of the colleges
in order to best allocate seats.

The Purpose of this Report

We created a web application that helps a candidate to fill his/her choices for JEE counselling. We also created a program in java that takes into account the preferences of students and the merit lists of the colleges
in order to best allocate seats.
This report is made through beamer which is a very useful tool. It is much more powerful than other presentation softwares such as MS Power Point.

The Purpose of this Report

We created a web application that helps a candidate to fill his/her choices for JEE counselling. We also created a program in java that takes into account the preferences of students and the merit lists of the colleges
in order to best allocate seats.
This report is made through beamer which is a very useful tool. It is much more powerful than other presentation softwares such as MS Power Point. There are many predefined templates , which can be applied very easily, and make the presentation attractive and interactive.

The Purpose of this Report

We created a web application that helps a candidate to fill his/her choices for JEE counselling. We also created a program in java that takes into account the preferences of students and the merit lists of the colleges
in order to best allocate seats.
This report is made through beamer which is a very useful tool. It is much more powerful than other presentation softwares such as MS Power Point. There are many predefined templates , which can be applied very easily, and make the presentation attractive and interactive.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.
Those waitlisted are selected for that course and others don't get a seat.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.
Those waitlisted are selected for that course and others don't get a seat.
This algorithm manipulates the preference list of candidates and the merit list in such a way that it automatically takes care of de-reservation.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.
Those waitlisted are selected for that course and others don't get a seat.
This algorithm manipulates the preference list of candidates and the merit list in such a way that it automatically takes care of de-reservation.
Each choice is split according to candidates category in the preference list.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.
Those waitlisted are selected for that course and others don't get a seat.
This algorithm manipulates the preference list of candidates and the merit list in such a way that it automatically takes care of de-reservation.
Each choice is split according to candidates category in the preference list.
Also each program gets its own modified meritlist based on programs category.

Part 1a: Gale-Shapley Algorithm

In this algo all the candidates apply simultaneously. [3]
On second iteration, those waitlisted apply again to the same program and those rejected apply to the next.
This keeps happening untill all are either waitlisted or out of choices.
Those waitlisted are selected for that course and others don't get a seat.
This algorithm manipulates the preference list of candidates and the merit list in such a way that it automatically takes care of de-reservation.
Each choice is split according to candidates category in the preference list.
Also each program gets its own modified meritlist based on programs category.
Due to these changes we take care of de-reservation in one go of algorithm.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.
Those who are leftout now apply according to their category in the main algorithm.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.
Those who are leftout now apply according to their category in the main algorithm.
For foreign candidates, we take care of them right at the end.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.

Those who are leftout now apply according to their category in the main algorithm.

For foreign candidates, we take care of them right at the end.

Their preference list is also manipulated to take care of de-reserved seat.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.
Those who are leftout now apply according to their category in the main algorithm.
For foreign candidates, we take care of them right at the end.
Their preference list is also manipulated to take care of de-reserved seat.
After all this, we get an optimal and fair allocation.

Part 1b: Gale-Shapley Algorithm

In order to accomodate DS students they are given institute seats before the start of algorithm.
Those who are leftout now apply according to their category in the main algorithm.
For foreign candidates, we take care of them right at the end.
Their preference list is also manipulated to take care of de-reserved seat.
After all this, we get an optimal and fair allocation.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.
We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.
We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.
Once a candidate gets a seat it is fixed for him and he cannot be replaced.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.
We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.
Once a candidate gets a seat it is fixed for him and he cannot be replaced.
If a candidate appears multiple times in the CML then he gets multiple chances to go through his preferences unless he already got selected.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.
We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.
Once a candidate gets a seat it is fixed for him and he cannot be replaced.
If a candidate appears multiple times in the CML then he gets multiple chances to go through his preferences unless he already got selected.
After one go of the algorithm we take care of de-reservation i.e vacant seats are added to GE, SC and ST according to rule.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order

are appended in the same order.

We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.

Once a candidate gets a seat it is fixed for him and he cannot be replaced.

If a candidate appears multiple times in the CML then he gets multiple chances to go through his preferences unless he already got selected.

After one go of the algorithm we take care of de-reservation i.e vacant seats are added to GE, SC and ST according to rule.

After this we go through the list again as earlier.

Part 2a: Merit-Order Algorithm

In this algo we make a common meritlist in which GE, OBC, SC, ST, GE-PD, OBC-PD, SC-PD and ST-PD meritlists all present in increasing order
are appended in the same order.
We go through the list and each candidate(except foreign) gets a chance to apply in its preference list untill he gets an empty seat or his list is exhausted.
Once a candidate gets a seat it is fixed for him and he cannot be replaced.
If a candidate appears multiple times in the CML then he gets multiple chances to go through his preferences unless he already got selected.
After one go of the algorithm we take care of de-reservation i.e vacant seats are added to GE, SC and ST according to rule.
After this we go through the list again as earlier.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.
After this round all those who were not selected will apply again along with others but this time according to their category.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.
After this round all those who were not selected will apply again along with others but this time according to their category.
For foreign candidates, they apply at the end.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.

After this round all those who were not selected will apply again along with others but this time according to their category.

For foreign candidates, they apply at the end.

We go through their list in the same way as others and there is no need to worry about de-reservation as it has already taken place earlier.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.
After this round all those who were not selected will apply again along with others but this time according to their category.
For foreign candidates, they apply at the end.
We go through their list in the same way as others and there is no need to worry about de-reservation as it has already taken place earlier.
After all this we get an allocation which may not be fair for some cases.

Part 2ab: Merit-Order Algorithm

In order to take care of DS category we go through the DS list and apply the same algo for institute seats in place of programs.

After this round all those who were not selected will apply again along with others but this time according to their category.

For foreign candidates, they apply at the end.

We go through their list in the same way as others and there is no need to worry about de-reservation as it has already taken place earlier.

After all this we get an allocation which may not be fair for some cases.

Part 3: Choice Filling Web Application

- We designed a JEE choice filling Web application which lets a student fill his/her choices for JEE counselling

Part 3: Choice Filling Web Application

- 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 [1]

Part 3: Choice Filling Web Application

- 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 [1]
- In this project report, we have tried to elaborate our project work

Part 3: Choice Filling Web Application

- 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 [1]
- 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 [2]

Part 3: Choice Filling Web Application

- 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 [1]
- 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 [2]

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

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

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

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

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

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

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

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

Conclusion

We did enjoy a lot, and learning Java and Django was a great experience in itself

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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,

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

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

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

References

DjangoBook.
Django Tutorial.
http://www.djangobook.com/en/2.0/index.html.

DjangoProject.
Django Tutorial.
https://www.djangoproject.com/).

Tutorials Point.
Tutorials for learning Java.
http://www.tutorialspoint.com/java/.