



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SESSION 2023/2024
SEMESTER 1

SECJ2013 - DATA
STRUCTURE & ALGORITHM

Assignment

Assignment 1 : Sorting & Searching

Group Name : Agent P

Lecturer: Madam . Lizawati binti Mi Yusof

GROUP MEMBERS:

No.	Name	Matric No.
1	NAVINDRAN A/L RAGHUPATHY	A22EC0227
2	KUGHANRAJ A/L ARUNASALAM	A22EC0179
3	KUGANES VARMAN A/L BALAN	A22EC0176

Table of Contents

1.0 Introduction.....	3
2.0 Objectives.....	3
3.0 Synopsis.....	3
4.0 Design.....	4
4.1 Class Diagram.....	4
4.2 Flowchart.....	5
5.0 Sorting & Searching.....	6
5.1 Sorting.....	6
5.1.1 Sorting by Name.....	6
5.1.2 Sorting by Faculty.....	6
5.1.3 Sorting by IC.....	6
5.1.4 Sorting by Age.....	6
5.1.5 Sorting by Matric Number.....	7
5.1.6 Sorting by Vote Count.....	7
5.2 Searching.....	7
5.2.1 Search for name.....	7
5.2.2 Search for faculty.....	7
5.2.3 Search for IC.....	8
5.2.4 Search for Age.....	8
5.2.5 Search for Matric Num.....	8
5.2.6 Search for Vote Count.....	8

1.0 Introduction

As from the agent P group we decided to develop an election management system for UTM. From this system UTM can handle the elections easily and efficiently. For assignment 1 we had developed a class for candidates which contain 7 attributes which define a candidate. The attributes are name , age , identification number , phone number , vote count , faculty and matric number.

2.0 Objectives

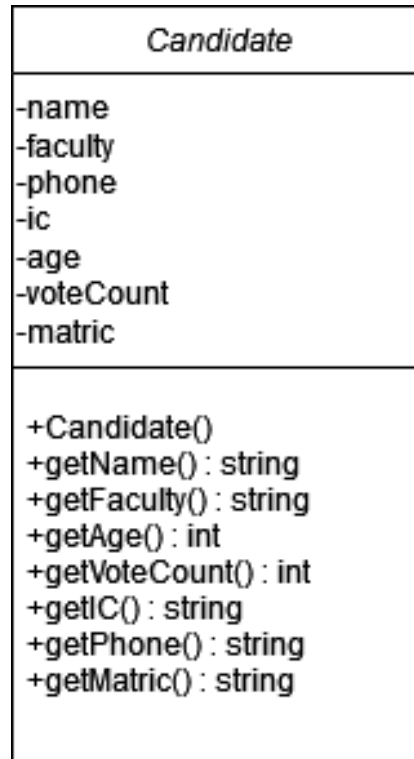
The objective for this assignment 1 is to create an Election management system. In this system it will enable users to get a sorted version of the Election candidate list by attributes like name, faculty, age, vote count, IC and matric num. Each attributes stated will be sorted in ascending and descending order where user can choose which way they want to view the candidate list. The system will also enable users to do searching on the stated attributes. If the user wanted to get the detail of a candidate by entering any of the attributes value, the system will display the candidate's details. Thus, the aim of this system is to create a user friendly interface where user can easily use the election management system.

3.0 Synopsis

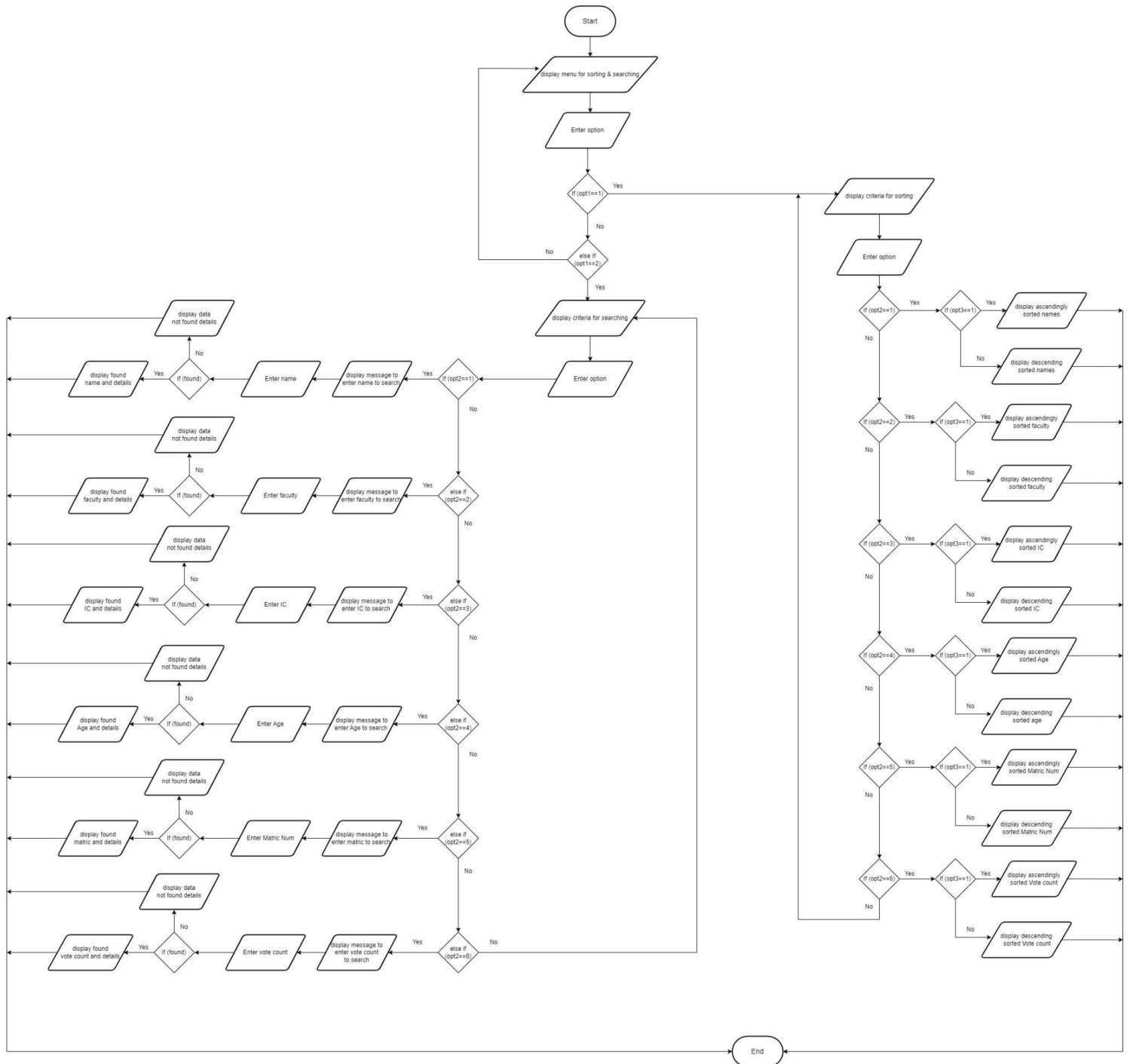
Election Management System is a system that can help students in UTM to view the candidates information like name, faculty, age, matric number, phone number and vote count. There is also a sorting and searching algorithm implemented in the system so that students have an easier time looking for the candidates. For example, students can search for the candidates from their faculty so that they can support the candidate from their faculty. There is also sorting that will arrange the candidates in ascending or descending order depending on the information they want to sort. For example, students can see the candidates from oldest to youngest or vice versa. This system will ease students in their search for candidates as all candidates are registered in the system.

4.0 Design

4.1 Class Diagram



4.2 Flowchart



5.0 Sorting & Searching

5.1 Sorting

5.1.1 Sorting by Name

For sorting by name, we have implemented the selection sorting algorithm. There are two functions for sorting by name which are sorting in ascending order and descending order. The selection sorting function accepts two parameters which are object array of candidate class and size of the array. In ascending order the names are sorted from A-Z while in descending order the names are sorted from Z-A.

5.1.2 Sorting by Faculty

For sorting by faculty, we have implemented the improved bubble sorting algorithm. There are two functions for sorting by faculty which are sorting in ascending order and descending order. The improved bubble sorting function accepts two parameters which are object array of candidate class and size of the array. In ascending order the faculties are sorted from A-Z while in descending order the faculties are sorted from Z-A.

5.1.3 Sorting by IC

Sorting by IC is implemented using selection sorting algorithm. There are two functions for this sorting by IC, which are in ascending order and descending order. Both function has two parameters which are the object array of candidate class and size of the array. In ascending order the candidates are arranged according to their IC values where they are sorted from the smallest IC value to the largest IC value. The descending is the opposite of the ascending order where the candidate will be sorted from largest IC value to the biggest IC value.

5.1.4 Sorting by Age

For sorting by Age, we implemented insertion sorting. There are two functions for this sort, one for ascending order and another one for descending order. The algorithm will compare the candidates age (int) and arrange it following the ascending or descending order.

5.1.5 Sorting by Matric Number

For sorting by Matric Number, we implemented selection sorting. There are two functions for this sort, one for ascending order and another one for descending order. The algorithm will compare the candidate's Matric number (string) and arrange it following the ascending or descending order.

5.1.6 Sorting by Vote Count

Sorting by vote count is implemented using insertion sorting algorithm. There are two functions for this sorting by vote count, which are in ascending order and descending order. Both functions have two parameters which are the object array of candidate class and size of the array. In ascending order the candidates are arranged according to their vote counts where they are sorted from the smallest vote counts to the largest vote count. The descending is the opposite of the ascending order where the candidate will be sorted from largest vote counts to the biggest IC value.

5.2 Searching

5.2.1 Search for name

For searching by name, we have implemented sequential search to get the details of the candidate based on the name entered by the user. We have implemented the sequential in this searching function because it can be used in both sorted and unsorted lists. If the entered key was successfully found in the list it will display the details to the corresponding name. If the data not found in the array it will display the message "Data not found".

5.2.2 Search for faculty

For searching by name, we have implemented sequential search to get the details of the candidate based on the name entered by the user. We have implemented the sequential in this searching function because it can be used in both sorted and unsorted lists. If the entered key was successfully found in the list it will display the details to the corresponding name. If the data not found in the array it will display the message "Data not found".

2.2.3 Search for IC

We have implemented sequential search for searching by IC to get the candidate's detail based on the IC entered by the user. User will be asked to enter the IC of the candidate they want to search for and if the function found a match, it will display the candidate's information, but if the IC does not match with any of the candidate's IC, it will display "Data entered not found!".

5.2.4 Search for Age

Students will be asked to enter the Age they want to search for and the function will display the candidate(s) and if the age they searched for was not found, a message will be displayed.

5.2.5 Search for Matric Num

Students will be asked to enter the Matric number they want to search for and the function will display the candidate and if the Matric number they searched for was not found, a message will be displayed.

5.2.6 Search for Vote Count

We have implemented sequential search for searching by vote count to get the candidate's detail based on the IC entered by the user. User will be asked to enter the vote count of the candidate they want to search for and if the function found a match, it will display the candidate's information, but if the vote count does not match with any of the candidate's IC, it will display "Data entered not found!".