

# ITU Computer Engineering Department

## BLG 223E Data Structures, Spring 2023

### Homework 1

April 1, 2023

## 1 Introduction

In this assignment, you implement a multi-level sorted linked list and array data structure by using the YÖK Atlas data of the Yüksek Öğretim Kurumu(YÖK is an institution that manages, regulates and supervises higher education institutions in Turkey.), which includes the statistics of students placed in all departments of universities in Turkey. You will compare the execution times of the operations you will do on these two different data structures. You will prepare a report comparing execution times.

## 2 Dataset

You will work on the university data file in the YÖK Atlas system [1]. There are 12 columns: Department name, Faculty name, University name, Scholarship amount(Discount), Language, Last settled person's score, Last settled person's rank, First person's score to settle, First person's rank to settle, Average rank of settled, Average score of settled. The data is in csv (Comma-separated values) format. A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields. The CSV file format is not fully standardized. Separating fields with commas is the foundation, but commas in the data or embedded line breaks have to be handled specially. The semicolon(;) is used as a separator in the data given in the assignment. Your separator may be different depending on your operating system or the spreadsheet you are using. A sample code is provided to read the csv file. Update your separator in this code by examining the given data. As the data is translated from Turkish to English, there may be some Turkish words in between. You are not obligated to translate them. You do not need to modify the data.

## 3 Problem

You will implement two data structures in which the departments are sorted alphabetically first, and secondly, the universities belonging to each department will be sorted alphabetically according to their university names. You will use 2d multilist and array data structures for this. You can see the final sorted multilevel linked list in the Figure 5 and sorted array in the Figure 6.

### 3.1 2D Linked List Solution

You create a 2D sorted linked list. 3 classes will be created (Department, University, MultiList). The first dimension will hold departments(Department Node), the second dimension will hold universities(University Node). You will submit 5 files: MultiNode.h, MultiNode.cpp, MultiList.h, MultiList.cpp and main.cpp.

Each horizontal node in a multilevel linked list has next, previous and up pointer and each vertical node has up pointer. These nodes are represented as:

### Code Listing 1: MultiNode.h

```
class University{
private:
    string faculty;
    string discount;
    string city;
    string uniName;
    string lang;
    string score_last;
    string range_last;
    string score_first;
    string range_first;
    string average_range;
    string average_score;
    University* up;

public:
    University(string, string, string, string, string, string,
    string, string, string, string, string);
    University* getUp();
    void set_up(University*);
    string get_uni();
    void printDetails();
};

class Department{
private:
    string department;
    int numOfUniversity;
    Department* next;
    Department* prev;
    University* under;

public:
    Department(string);
    Department* getNext();
    Department* getPrev();
    string get_Department();
    int get_numOfUni();
    University* getunder();
    void increase();
    void set_next(Department* );
    void set_up(University*);
    void set_prev(Department*);
};
```

You will write the functions of each class in the MultiNode.cpp file. Include the MultiNode.h file in MultiNode.cpp.

*Details of MultiNode.h and MultiNode.cpp files:*

- Write constructor functions of each class.
- up, next and prev pointers must be private.
- Since these attributes are private, write their functions to get and set them.
- Print the details of the university with printDetails() function. The function should print the faculty, discount, city, language, last score, last rank, first score, first rank, average score, average rank of the university (This function belongs to University class).
- You will keep data on how many universities each department contains in the Department class with numOfUniversity attribute. You can write increase() function in each university adding to department operation.
- **The list must be sorted horizontally by department and vertically by university name (alphabetical).** For universities with the same name, the order in the list does not matter.

- You can add different functions (if necessary).
- You can use different naming for attributes and functions.

To implement multilevel linked list write MultiList.h and MultiList.cpp files. Multilevel linked list is represented as:

#### Code Listing 2: MultiList.h

```
class MultiList{
private:
    Department* head;
    Department* tail;
public:
    MultiList();
    ~MultiList();
    void add_node(string, string, string, string, string, string, string, string, string, string, string, string, string);
    void add_uni_node(Department*,string, string, string, string, string, string, string, string, string, string, string);
    void add_dep_uni_node(Department*,string,string, string, string, string, string, string, string, string, string, string);
    void select_department();
    void select_uni();
    void listDepartments();
};
```

You will write the functions of MultiList class in the MultiList.cpp file. Include the MultiNode.h file in MultiList.cpp. And include MultiList.h file in the MultiList.cpp file

*Details of MultiNode.h and MultiNode.cpp files:*

- Write constructor and destructor of the class.
- head and tail attributes must be private.
- You are taken university information from the file. You add this information to sorted multilevel linked list by department name and university name. If the department node is already created, you should not create a department horizontally and add this data up to this node by university. You can increase or decrease the number of functions to perform adding node operation. **Departments and universities will be added to multilevel linked list alphabetically. Sorting will not be done after adding to the list.**
- select\_department() function will ask the user which of the departments in alphabetical order she/he wants to see. The user will be prompted to enter a number between 1 and the department number. You can define the number of departments globally or pass it as a parameter to the function. If the user enters a number that is more than the number of departments, ask to enter the number again. Then print all the universities of that department.
- select\_uni() function will ask the user which of the departments in alphabetical order she/he wants to see. The user will be prompted to enter a number between 1 and the department number. You can define the number of departments globally or pass it as a parameter to the function. If the user enters a number that is more than the number of departments, ask to enter the number again. Print which department the number entered by the user belongs to. Again, ask the user to enter a number to see the information of a university in the selected department, which are listed alphabetically. Specify that the user must enter a number in the range of 1 and university number. If the user enters a number out of range, ask to enter the number again. Print the information of the university to which the number entered corresponds to the screen. An example output can be seen in the below. You must change the department number and university number texts with the department number integer values.

#### Example Output of select\_uni() Function

To review a department, enter a number between 0 and the department number: 1  
Department: Accounting and Finance Management

```

To review a university of selected department, enter a number between 0 and the
university number: 2
Baskent University
Faculty: Commercial Sciences Faculty
Discount: 50% Discount
City: ANKARA
Language: Turkish
Score Last: 233
Range Last: 507990
Score First: 233
Range First: 507990
Score Average: 217
Range Average: 670189

```

- `listDepartments()` function prints all the department names and number of universities in each department. A part of the output can be seen in the below.

### A Part of Output of the `listDepartments()` Function

```

Department: Photonics
Number of Universities of Department: 4
-----
Department: Physical Education and Sports Teaching
Number of Universities of Department: 31
-----
Department: Physical Therapy and Rehabilitation
Number of Universities of Department: 470
-----
Department: Physics
Number of Universities of Department: 161
-----
Department: Physics Engineering
Number of Universities of Department: 13

```

## 3.2 Array Solution

Define a 2-dimensional array. Allocate the array memory dynamically. Include the following information on each row: Department name, Faculty name, University name, Scholarship amount(Discount), Language, Last settled person's score, Last settled person's rank, First person's score to settle, First person's rank to settle, Average rank of settled, Average score of settled.

Array must first be sorted alphabetically by department name. Universities within each department should also be in alphabetical order by university name (as in the multi linked list). Move all data to next row each time you add new data to the array. You can calculate the number of departments and the number of universities for each department while filling the array. You can then pass them as parameters to the functions. Do not allow duplication when printing all departments. De-allocate all the memory you dynamically set for the array before you finish the program.

## 3.3 Execution Time Measuring

Print the execution times to the terminal to compare these two data structures. You can add the `chrono` library for this. Check the link for detailed explanation: <https://www.geeksforgeeks.org/measure-execution-time-function-cpp/>. The processes for which execution times will be printed are:

- Fill List/Array - Run counter after creating list/array. Stop the counter after adding all the data to the list/array. Print the creation and filling time of the data structure to the terminal by subtracting the start point of the counter from the end point.
- List All Departments - Start the counter when calling the function and stop the counter when all departments are printed.
- List All Universities - Start the counter when calling the function and stop the counter when all universities are printed.

- Select a Department and University - Start the counter when calling the function and stop the counter when the information about the selected university with selected department is printed.
- De-allocate List/Array - Start the counter when starting to de-allocate the memory and stop the counter when the list/array is deleted.

After measuring these times, comment them and compare the execution times in the report.

## 4 Example Screenshots

These operations must be performed on both data structures.

`print_menu()` is given. Write `perform_operation()` function to get a choose from the user (The user must be able to use lowercase or uppercase letters).

L character is used to list all departments and number of universities of that department. You can see an example screenshot of the L operation in the Figure1.

D character is used to select a department to list all universities with its information of that department. **DO NOT forget to change department number text with the integer value(e.g. 200).** See in Figure 2.

U character is used to print the information of a university selected by user with number. **NOTE: The user selects the department number first. DO NOT forget to change department number and university number text with the integer values(e.g. 200).** See in Figure 3.

E character is used to exit. And also add the feature that if user enters anything other than these four letters, prompt them to make an error and make a selection again. See in Figure 4.

Your multilevel linked list should be as in the figure 5.

Your array should be as in the figure 6.

## 5 Report

Write a short report (1-2 pages) about your homework. Specify the desired execution times in [section 3.3](#) for both data structures in the report. Then compare the times. For example, in these two data structures, specify that insertion is faster in the list. Mention the advantages and disadvantages of both data structures.

## 6 Submission Rules

- You cannot use Standard Template Library (STL). You must write all the list operations yourself. Do not use libraries other than those specified.
- Make sure you write your name and number in all of the files of your project, in the following format:  

```
/* @Author
Student Name: <student name>
Student ID : <student id>
Date: <date> */
```
- You will submit 5 files for multi-list list in one zip. For array implementation you will submit one zip. Then you will submit a PDF for report.
- Use comments wherever necessary in your code to explain what you did.
- Your program will be checked by using Calico (<https://bitbucket.org/uyar/calico>) automatic checker.
- Do not share any code or text that can be submitted as a part of an assignment (discussing ideas is okay).
- Only electronic submissions through Ninova will be accepted no later than deadline.

- You may discuss the problems at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your **own, individual homework**.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- If you have any question about the recitation, you can send e-mail to İlknur Çelik (celikil17@itu.edu.tr).
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**

## References

- [1] Yok atlas. <https://yokatlas.yok.gov.tr/lisans-anasayfa.php>. Accessed: 2023-03-10.

Figure 1: List All Departments

```
Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: L
Department: Accounting and Auditing
Number of Universities of Department: 6
-----
Department: Accounting and Finance Management
Number of Universities of Department: 49
-----
Department: Actuarial Sciences
Number of Universities of Department: 20
-----
Department: Advertising
Number of Universities of Department: 52
-----
Department: Advertising Design and Communication
Number of Universities of Department: 9
-----
Department: Aeronautical Engineering
Number of Universities of Department: 37
-----
Department: Aerospace Engineering
Number of Universities of Department: 22
-----
Department: Agricultural Biotechnology
Number of Universities of Department: 27
-----
Department: Agricultural Economics
Number of Universities of Department: 67
-----
Department: Agricultural Engineering Programs
Number of Universities of Department: 3
-----
Department: Agricultural Genetic Engineering
Number of Universities of Department: 3
-----
Department: Agricultural Machinery and Technologies Engineering
Number of Universities of Department: 30
-----
Department: Agricultural Structures and Irrigation
Number of Universities of Department: 20
-----
Department: Agricultural Trade and Management
Number of Universities of Department: 6
-----
Department: Aircraft Body and Engine Maintenance
Number of Universities of Department: 20
-----
```

Figure 2: Select A Department

```

Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: d

To review a department, enter a number between 0 and the department number: 1
Department: Accounting and Finance Management
1: Afyon Kocatepe University
Faculty: Bolvadin Applied Sciences Faculty
Discount: Free
City: AFYONKARAHISAR
Language: Turkish
Score Last: 230
Range Last: 529147
Score First: 230
Range First: 529147
Score Average: 262
Range Average: 776142
-----
2: Baskent University
Faculty: Commercial Sciences Faculty
Discount: 50% Discount
City: ANKARA
Language: Turkish
Score Last: 210
Range Last: 965977
Score First: 258
Range First: 527217
Score Average: 267
Range Average: 321165
-----
3: Baskent University
Faculty: Commercial Sciences Faculty
Discount: 50% Discount
City: ANKARA
Language: Turkish
Score Last: 233
Range Last: 507990
Score First: 233
Range First: 507990
Score Average: 217
Range Average: 670189
-----
4: Baskent University
Faculty: Commercial Sciences Faculty
Discount: 100% Discount
City: ANKARA
Language: Turkish
Score Last: 256
Range Last: 343508

```

Figure 3: Select A Department and University

```

Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: u

To review a department, enter a number between 0 and the department number: 1
Department: Accounting and Finance Management

To review a university of selected department, enter a number between 0 and the university number: 2
Baskent University
Faculty: Commercial Sciences Faculty
Discount: 50% Discount
City: ANKARA
Language: Turkish
Score Last: 233
Range Last: 507990
Score First: 233
Range First: 507990
Score Average: 217
Range Average: 670189
-----
Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: 

```



Figure 4: Exit

```

Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: r
ERROR: You have entered an invalid choice

Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: t
ERROR: You have entered an invalid choice

Choose an operation
L: List All Departments
D: Select a Department
U: Select a Department and University
E: Exit
Enter a choice {L,D,U,E}: e

```

Figure 5: Sorted multilevel linked list.

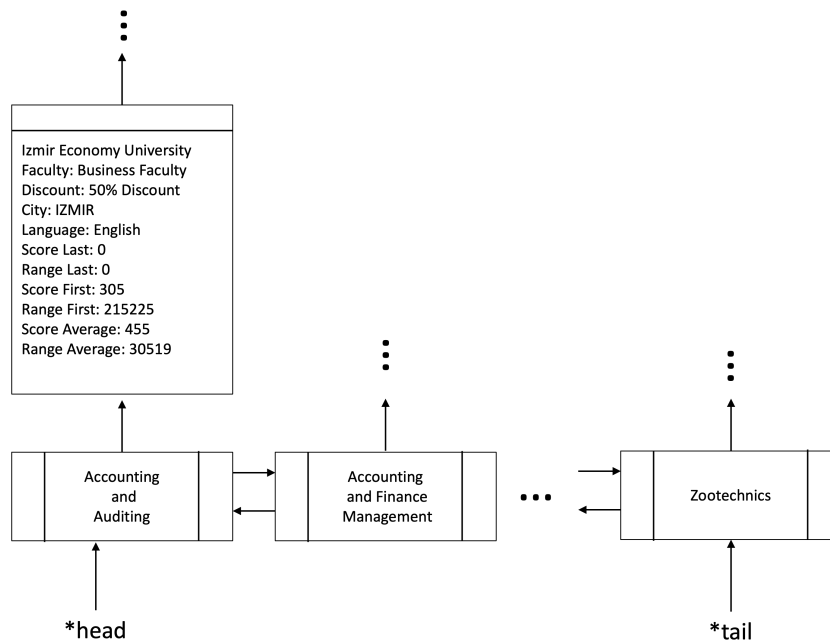


Figure 6: Sorted array.

Accounting and Auditing	Business Faculty	Izmir Economy University	50% Discount	IZMIR	English	0	0	305	215225	30519	455
Accounting and Auditing	Business Faculty	Izmir Economy University	100% Discount	IZMIR	English	356	85606	358	80830	39709	448
Accounting and Auditing	Business Faculty	Izmir Economy University	50% Discount	IZMIR	English	234	719256	276	408871	264579	332
<div style="text-align: center;"> <p>•</p> <p>•</p> <p>•</p> </div>											
Zootechnics	Agriculture Faculty	Van Yuzuncu Yil University	Free	VAN	Turkce	224	588217	276	321666	299385	268
Zootechnics	Agriculture Faculty	Van Yuzuncu Yil University	Free	VAN	Turkce	230	368637	259	262937	711384	259