EE441- Programming Assignment 1

Due Date: 11.11.2024, 23:59

For your questions: Utkucan Doğan – utkucan@metu.edu.tr

This assignment consists of one part. You are going to create a makefile project for this part. You are also going to prepare a pdf file with your answers to some questions. Don't forget to write comments to your code as they are also graded. **Follow the naming conventions given in the boxes, otherwise your answer will not be graded.**

Sorted List [100 points]

You are going to implement a sorted list class which has its elements always be sorted. This will be held when a new element is added, or an existing element is removed. Follow the given steps to complete this question. Use `<stdexcept>` library for the exceptions.

1) **[5 points]** Implement a class named `SortedList`. This class should have a `float` array to hold the elements and a `size_t` for the size of the list as its private members. Define a macro `SORTEDLIST_MAX_SIZE` for array size and make its value 20.

2) **[5 points]** Implement a default constructor which will initialize the size of the list as zero.

3) **[5 points]** Define a public member function which copies a given list into the object.

```
void copy(const SortedList& other);
```

4) **[5 points]** Define a public member function which will return the number at the given index. Function should throw an `std::out_of_range` exception if the index is out of range.

```
float index(size_t ind);
```

5) **[15 points]** Define a public member function that inserts the given number into the correct place and returns its index. This function should throw `std::length_error` exception if the insertion will exceed the maximum array size. Follow the steps given below.

```
size_t insert(float number);
```

a) Check if there is enough space in the list. Throw the error if there isn't.
b) Find the correct index by employing a linear search.
c) Update the size and shift the existing values in the list to have a space for the new value.
d) Insert the value into the list and return its index.

6) **[15 points]** What is the time complexity of the insert function? Will the complexity change if a binary search is used instead of a linear search? Show your calculations.

7) **[10 points]** Define a public member function that removes the number in the given index and returns it. This function should throw `std::out_of_range` exception if the index is out of range. Follow the steps given below.

```
float remove(size_t index);
```

a) Check if the index is in range. Throw the error if it isn't.
b) Put the value in the index into a temporary variable.
c) Shift the values in the list to fill the empty space.
d) Update the size and return the value.

8) **[10 points]** What is the time complexity of the remove function? Show your calculations.

9) **[10 points]** Define a public member function that finds the index of a given number in the list with a binary search. This function should throw `std::domain_error` exception if the number is not in the list.

```
size_t find(float number);
```

10) **[10 points]** What is the time complexity of the find function? Show your calculations.

11) **[5 points]** Define a public member function that prints the values in the list with a space between each value. If the list is empty this function should print "The list is empty".

12) **[5 points]** Define a main function that demonstrates the functionality of your class. It should show all the criteria expected in the questions.