

Lab 5 – More List Functions

Adding functionality to Array Lists and Linked Lists

Task 1. Add functionality to the **arrayList** template:

Add the following functionalities to the **arrayList** template from Lab 3:

Function	Description
void extend(const arrayList *)	Extend the arrayList with the elements in the arrayList passed in the first parameter
elemType min() const	Returns the minimum element in the list
elemType max() const	Returns the maximum element in the list
int count(const elemType&)	Counts the number of elements in the list matching the element passed in the first parameter

Here, **elemType** is the type of the members of the list. In a given list, all elements are of the same type. You should use [template](#) implementation to enable functionality to have lists storing different types of objects dynamically.

Task 2. Add functionality to the **linkedList** template:

Add the following functionalities to the **linkedList** template from Lab 4:

Function	Description
void extend(const arrayList *)	Extend the arrayList with the elements in the arrayList passed in the first parameter
elemType min() const	Returns the minimum element in the list
elemType max() const	Returns the maximum element in the list
int count(const elemType&)	Counts the number of elements in the list matching the element passed in the first parameter

Here, **elemType** is the type of the members of the list. In a given list, all elements are of the same type. You should use [template](#) implementation to enable functionality to have lists storing different types of objects dynamically.

Task 3. Algorithm Analysis, Thoughts and Documentation:

Discuss the following:

1. Compute the time complexity of the four functions in task 1, using big-oh notation.
2. Compute the time complexity of the four functions in task 2, using big-oh notation.

Specifications

Below is a breakdown of the tasks labeled [LP] and [HP]. If you complete ALL the LP components satisfactorily, you will receive a grade of “low pass” on the lab. If you complete ALL the LP components and the HP components mentioned below satisfactorily, you will receive a grade of “high pass”:

- **[LP]** Task 1 functionalities implemented as a template using dynamic memory management correctly.

- [LP] All task 1 implemented functionalities are tested in main.
- [LP] Task 2 functionalities implemented as a template using dynamic memory management correctly.
- [LP] All task 2 implemented functionalities are tested in main.
- [HP] Task 3 completed satisfactorily.

If you do not meet the criteria for a “low pass”, the submission will be marked as “revision needed”.

What to submit:

Your final submission will need to have the files as follows:

- `arrayList.h`
- `linkedList.h`
- Notes (if you complete the optional HP task)
- `lab5-cmpe126.cpp`

NOTE: You can look for help on the Internet but refrain from referencing too much. Please cite all your sources in your Notes file.

When to submit:

Submit your lab before **Thursday, March 21st, 11:59pm**. You are strongly advised to submit before Friday, March 15th, 11:59pm.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to provide the reason.

“I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file.”