

Lab 8 – Queues

Implement queues using various underlying data structures

Task 1. [LP] Implement queues using a circular array

Create a template called `arrayQueue` that implements a queue using an array in a file called `arrayQueue.h`. The template should have the functionalities of `enqueue`, `dequeue`, `front`, `size`, `maxSize`, `isEmpty`, and `isFull` as described in the lecture. Remember to make sure that the array is circular, i.e., the queue wraps around the end of the array. Test the functionality of your queue in a file called `main.cpp`. A sample `main.cpp` is provided to you. You are required to edit it and test for other cases as well. You may use STL templates for the array.

Task 2. [HP] Implement queues using two stacks

Implement a queue using two stacks.

1. Create a template

```
template <typename elemType>
class stackQueue {...1};
```

in a file called `stackQueue.h` with two fields of type `stack` (you can use the STL stack here. If you want to use your own implementation, that is fine too). Other fields should not store queue elements. Note that the stacks can only support push and pop operations to add/remove items from the data structure. You may use auxiliary data structures within the member functions.
2. The template should have the functionalities of `enqueue`, `dequeue`, `front`, `size`, `maxSize`, `isEmpty`, and `isFull` as described in the lecture.
3. Write the pseudocode and the time complexity in big-O notation for each of the above operations in a file called `stackQueue.pdf`. Please explain your design choices as well.
Note that you may refer to online sources for help but write the pseudocode on your own. That will help avoid getting into trouble for violating the honesty pledge and to make sure you understand the algorithm well.
4. Implement the pseudocode in `stackQueue.h`.
5. Test your code using `main.cpp`. A sample `main.cpp` is provided to you. You are required to edit it and test for other cases as well.

¹ Template declaration goes here.

Specifications

Task 1 is labeled [LP] and task 2 is labeled [HP]. If you complete task 1 satisfactorily, you will receive a grade of “low pass” on the lab. If you complete both tasks satisfactorily, you will receive a grade of “high pass”. 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:

- `arrayQueue.h`
- `main.cpp`
- Notes to include all your references
- `stackQueue.h` if you are implementing task 2

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 **Monday, May 6th, 11:59pm**. You are strongly advised to submit before Friday, April 26th, 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.”