

Assignment 2

CSL 201 Data Structures, Sem 1, AY 17/18, IITRPR

40 marks

Due on 10th October, 11:59 PM

Objective

The objective is to learn to implement three abstract data types: (1) Stacks, (2) Queues, (3) Heaps.

Instructions (IMPORTANT)

1. You are to use C++ programming language to complete the assignment.
2. Provide a Makefile to compile your final code.
3. This is an individual assignment. You can have high level discussions with other students, though.
4. Include a "Readme.txt" file on how to compile and run the code.
5. Upload your submission to moodle by the due date and time. There will be 10% penalty for late submission for first three days. After that, your assignment will not be evaluated.
6. If any student asks for deadline extension, he or she will get 5% penalty.
7. I should be able to use the class you have implemented in my program the way we use STL classes. There will be a master program to test the classes. If the class is templated, declare it in the hpp file itself.

Problem 1 [10 marks]

Design an ADT for a two-color, double-stack ADT that consists of two stacks—one "red" and one "blue"—and has as its operations color-coded versions of the regular stack ADT operations. For example, this ADT should allow for both a red push operation and a blue push operation. Give an efficient implementation of this ADT using a single array whose capacity is set at some value N that is assumed to always be larger than the sizes of the red and blue stacks combined.

Implement a modified stack in `cStack.hpp`. The `cstack` interface is given in file `cStack.hpp` with the assignment.

Problem 2 [15 marks]

Implement all the functions of the deque ADT using one cStack described above. The deque interface is given in deque.hpp.

Problem 3 [15 Marks]

Implement a heap data structure using linked list-based implementation. The heap should have the following additional functions: BuildHeap (A, n): $O(n)$ heap construction method given an array of numbers, MergeHeap (H): Merge another heap with the given heap in $O(n)$ time where n is size of the bigger heap. The heap should store student records consisting of student name, branch, and grade. The priority order is defined over grades. The input will be given in a text file, records.txt (example file attached with the assignment).

The interface of the heap is given in heap.cpp.