**Project 1 due Feb. 13th**

1. Write a class that maintains the top ten scores for a game application, implementing

 the add and remove methods using a singly linked list instead of an array

.

1. Perform the previous project, but use a doubly linked list. Moreover, your implementation of remove (i) should make a fewest number of pointer hops to get to the game entry at index i.

**Project 2 Due Feb. 27th**

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 support both a

redPush operation and a bluePush 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.

**Project 3 due March 13th**

Implement the array list ADT by means of an extendable array used in a circular fashion , so that insertions and deletions at the beginning and insertion at the end of the array list run in constant time.(circular link list)

**Project 4 due March 27th**

Implement insertion, selection, merge and quick sorts. Perform a series of benchmarking tests to see which one is faster. Your tests should include sequences that are “random” as well as ”almost” sorted.

**Project 5 due April 10th**

Write a program that can input and display a person’s family tree.

**Project 6 May 1th**

1. Implement the graph ADT using the adjacency list structure.
2. Implement the graph ADT using the adjacency matrix structure