

Programming assignment 1. **Heaps and pointers.**

Given February 5, due February 19.

It is often useful to combine two data structures using pointers. This might give a way to combine the functionality of both component data structures. Suppose that each object, x , is located in data structure E and data structure F . These are *threaded* if there is a pointer from the location of x in E to its location in F and vice versa. It is possible that the data structures E and F contain only the keys or even just pointers to the objects x . Look at the various insertion sort code fragments on the web page to see how this can be done. The extreme in this direction would be for E and F to contain pointers to the objects x , and the record in E pointing to x would also have a pointer to the record in F that points to x , and vice versa. The complicated thing, that will lead to endless bugs if you are not careful, is that whenever you move a record, say, in F , you have to update the pointer in the corresponding record in E . I strongly suggest that you write a method (or procedure or function or subroutine) `exchange(i,j,E,F)` that exchanges the records in locations i and j in E and updates the pointers to records in F . If you do this right, you can call `exchange(i,j,F,E)` to exchange records i and j in F and update the pointers to records in E .

This assignment uses these ideas to implement an “FLP queue” (“first and last priority queue”). The operations are `insert`, `getMax`, `getMin`, `deleteMax`, and `deleteMin`. Use one heap to implement `getMin` and another for `getMax`. An object, x , will contain a character string representing the title of the book and an integer representing the rating. The heaps will consist of records, each of which has a pointer to an object, the key of the object (here, the rating), and the pointer to the record in the other heap. We want to be able to insert books and access the best and worst of them. The data and operations to be performed are in a plain text file next to this one.