

Exercise Session

1. Discuss <http://java.sun.com/javase/6/docs/api/java/util/PriorityQueue.html>
2. Consider an extension of the stack ADT supporting POP(), PUSH(O) and MULTIPOP(K) operations. The first two operations are standard. MULTIPOP(K) removes top k elements from the stack, or clears the stack if the stack contains less than k elements. Pop and push both run in constant time, while MULTIPOP runs in time proportional to the number of elements removed (i.e. minimum of the stack size and k).

What is the total running time of n arbitrary operations executed on such a stack, assuming the stack was initially empty?

Note: this exercise does not assume any implementation details of the stack (whether it is an array or a list, or anything else). If you find yourself discussing such details then you are on the wrong track.

3. GT Exercise C-2.27 p. 135
4. GT Exercise C-3.3 p. 213 (assume that elements with keys equal to the key of current node are always stored in the left subtree).

Problems For Self-Study

1. Consider Java's TreeSet API description at <http://java.sun.com/javase/6/docs/api/java/util/TreeSet.html>. The `iterator()` method returns an iterator traversing the elements of the collection in an ascending order. How is this implemented, assuming that the TreeSet class uses a balanced extension of BSTs.
2. GT Ex. R-3.1, R-3.4, R-3.5, R-3.6, p. 212
3. GT Ex. C-3.2 p. 213.

Hint: Compare with Quiz of Episode 25, on binary search.

Homework Assignment

GT Exercise R-3.2, R-3.3 p. 212 and C-3.13 p. 214 [ca. 2-3 hours]

Deadline: before the lecture on March 7.