

Worksheet 8

Overview

The workshop for Week 9 will begin with a tutorial on HeapSort.

Tutorial Questions

Question 8.1 Construct a max binary heap from the following keys.

- 8 7 16 10 5 13 5 11 15 12 1 17

1. Construct a max binary heap using the top-down approach (one-at-a-time).
2. Now construct a max binary heap from the same keys, using the bottom-up "heapify" method.
3. What is the complexity of each method? Did the time it took you to do the exercise on paper correlate (roughly) with the theoretical complexity?

Programming exercises

Programming 8.1 An adaptive sorting algorithm takes advantage of already existing order in a file.

For example, insertion sort is an adaptive sort, since it exhibits linear behaviour with sorted or nearly sorted files.

Standard mergesorts (top-down or bottom-up) are not adaptive; they take $O(n \log n)$ time even for sorted files. However, *natural* mergesort is adaptive.

In natural mergesort, existing "runs" in the data are used as the first units input into a bottom-up mergesort. For example, if the first ten items are already in order, and the next five items are already in order, then the first items to be dequeued and merged would be a "run" of size 10 and a "run" of size 5. No time would be wasted merging the first two elements, or the next two, etc.

Write code for natural mergesort. Note that this only affects the set-up phase, so this should be fairly easy to adapt from the linked list code you wrote in last week's worksheet (if you didn't do last week's programming exercise, now is a good opportunity).