Advanced Operations on Priority Queues

4-24-2002

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignments. Hopefully everyone got my e-mail about assignment #6 before I left.

Heap Build Operation

- Last class we talked about using a heap for a priority queue where we wer constantly adding and removing elements. What is we have all the data and want to get it all into a heap?
- We could just do this by adding elements to an empty heap one at a time. This is O(n log n). We can also work on an existing "tree" structure to get a heap.

Heap Build Continued

- Using this approach, we run through all of the elements that aren't leaves and have each one percolate down through the tree just like elements did when we were removing from the heap.
- This operation in O(n) and the real reason why is that most of the items are one away from a leaf so they only require 1 swap at most.

Heap Sort

- We can use a heap build operation to implement an efficient sort. Obviously we could start with an empty heap then add elements to it and remove them to get everything back in sorted order. This requires extra memory like a mergesort did.
- If we use the array passed in for the heap we can remove the memory overhead.

Analysis of Heap Sort

- The heap sort is done in two main steps. First we use heapBuild to convert the array into a properly ordered heap. This, we said, if O(n).
- Second we want to remove the elements one at a time. This requires n removes and each one has a cost of O(log n) so this process takes O(n log n) time. The coefficient is fairly small and we can do it in a loop that doesn't use recursion.

Code

- Let's now take a bit of time to write a PriorityQueue class using a heap.

Minute Essay

- We have now discussed 3 O(n log n) sorts. What are they? Give a brief description and explain which one you would want to use.
- The second reading day is a week from tomorrow. Keep that in mind as you finish up your assignments.