

Multi-merge

You need to be alert to (usually minor) changes that may be made to the assignment statement or to the guidelines after the assignment is first put up. Refresh this frame and re-read the assignment carefully before you make your final submission.

Assignment statement

In merge sort, the principal operation is to merge two list of sorted data, say integers.

Here is a small example of merging two lists of integers sorted in ascending order:

1, 2, 3, 4, 7, 9, 10, 11, 15, 25 \leftarrow merge
routine \leftarrow $\begin{matrix} \leftarrow 1, 4, 7, 11, 15 \\ \leftarrow 2, 3, 9, 10, 25 \end{matrix}$

We consider a practical generalisation, where we have k lists of blocks of m items so that the items (integers) are sorted within and across the blocks in each list.

Here is an example with $k = 2$ and $m = 4$:

1, 2, 3, 4, 7, 9, 10, 11, 15, 19, 21, 25, 27, 30, 33, 38 \leftarrow merge routine
using heaps \leftarrow 1, 4, 7, 11, 15, 19, 21, 30
2, 3, 9, 10, 25, 27, 33, 38

Note that with a higher value of k there would have been more lists.

There will be up to km elements in the heap.

It is necessary to merge the integers of these lists so that the resulting is sorted and also grouped in blocks of m items so that the items (integers) are sorted within and across the blocks. At a time all integers are to be transferred out a block in the input list.

This generalisation is considered to model reading and writing to disks where i/o is done in block of some given size, say m .

The merging is to be done using Fibonacci heaps and associated operations. Integers from a block should be used to first create a Fibonacci heap and then that should be merged into the main Fibonacci heap used for creating the output list.

After integers have been read from a block and introduced to the heap, the last (and largest) integer read from that block should be reduced to one less than the minimum element in the heap if that is larger than the last value written to the merged list, if any.

This is an artificial thing to do. It has been introduced so that the DeckKey operation of heaps is also applied.

With respect to the example above using heaps, suppose the block containing 1, 4, 7, 11 has been read first and then the block containing 2, 3, 9, 10 has been read. The last integer in this

block is 10 and the minimum element in the heap is 1. The value 0 ($=1-1$) has not been written to the merged list, so the 10 should now be reduced to 0 (using the DecKey operation).

Input generation and testing

The numbers in the lists are to be kept unique and they should be in the range 1..10000. Values of k and m should be chosen suitably. In particular, testing will be done with k in the range 10..15 and m in the range 100..500.

As the range of integers is given, this automatically sets an upperbound on the size of the input that can be generated. For the experiment reasonably large inputs should be generated.

Maintain count of necessary operations, so that amortised costs for various operations can be computed and reported. Report your findings in a report.

Marking guidelines

Assignment marking is to be done only **after** the deadline expires, as submissions gets blocked after the assignment is marked.

Implementation of <code>makeHeap()</code> for Fibonacci heaps	2
Implementation of <code>insertHeap(H, x)</code> for Fibonacci heaps	8
Implementation of <code>extractMinHeap(H)</code> for Fibonacci heaps	12
Implementation of <code>unionHeap(H1, H2)</code> for Fibonacci heaps	6
Implementation of <code>decreaseKeyHeap(H, x, k)</code> for Fibonacci heaps	6
Reporting of amortised costs of operations	6
<i>Total Marks</i>	40

In your report you should discuss the implementation of each binomial heap operation and its complexity.

Assignment submission

Use electronic submission via the [WBCM link](#)

You should keep submitting your incomplete assignment from time to time after making some progress, as you can submit any number of times before the deadline expires.

Warning

Cases of copying will be dealt with seriously and severely, with recommendation to the Dean to de-register the student from the course.