Worst-Case Efficient External-Memory Priority Queues*

Gerth Stølting Brodal†
Max-Planck-Institut für Informatik
Im Stadtwald
D-66123 Saarbrücken, Germany
E-mail: brodal@mpi-sb.mpg.de
Fax: +49 681 9325 199

Jyrki Katajainen‡
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen East, Denmark
E-mail: jyrki@diku.dk
Fax: +45 3532 1401

Abstract

A priority queue Q is a data structure that maintains a collection of elements, each element having an associated priority drawn from a totally ordered universe, under the operations INSERT, which inserts an element into Q, and DELETEMIN, which deletes an element with the minimum priority from Q. In this paper a priority-queue implementation is given which is efficient with respect to the number of block transfers or I/Os performed between the internal and external memories of a computer. Let B and M denote the respective capacity of a block and the internal memory measured in elements. The developed data structure handles any intermixed sequence of INSERT and DELETEMIN operations such that in every disjoint interval of B consecutive priority-queue operations at most $c\log_{M/B}\frac{N}{M}$ I/Os are performed, for some positive constant c. These I/Os are divided evenly among the operations: if $B \geq c\log_{M/B}\frac{N}{M}$, one I/O is necessary for every $B/(c\log_{M/B}\frac{N}{M})$th operation and if $B < c\log_{M/B}\frac{N}{M}$, $\frac{c}{B}\log_{M/B}\frac{N}{M}$ I/Os are performed per every operation. Moreover, every operation requires $O(\log_2 N)$ comparisons in the worst case. The best earlier solutions can only handle a sequence of S operations with $O(\sum_{i=1}^{S}\frac{1}{B}\log_{M/B}\frac{N_i}{M})$ I/Os, where N_i denotes the number of elements stored in the data structure prior to the ith operation, without giving any guarantee for the performance of the individual operations.

Keywords: Priority queue, external memory algorithm, worst-case complexity