

The goal of this project is to use the theoretical techniques we have discussed in the first part of the course to design new algorithms and data structures. The project should be done in the same groups as project 1. A report with the solutions is due on *Friday April 8, 2005*. Remember to argue for correctness and complexity of each of your solutions. The evaluation of the project will be part of the final grade.

Sorting

1. Design an I/O-efficient algorithm for removing duplicate from a multiset of N elements (you can not assume the range of the elements is known); The output from the algorithms should be the K distinct elements among the N input elements in sorted order, and the algorithm should run in $O\left(\max\left\{n \log_m n - \sum_{i=1}^K n_i \log_m N_i, n\right\}\right)$ I/Os, where N_i is the number of copies of the i 'th elements in the input set.

(Hint: Use merge-sort and remove duplicates as soon as they are found. Analyze the algorithm by considering how many of the N_i copies of an element can be present after j merge steps.)

Searching

2. Design a linear space external data structure for the problem of maintaining a set of intervals I , such that given a query point x the *number* of intervals in I containing x can be reported in $O(\log_B^2 N)$ I/Os. The structure should support insertion and deletion of intervals in $O(\log_B^2 N)$ I/Os amortized.

Distribution sweeping

3. Design an $O(\text{sort}(N))$ I/O algorithm for the following problem using distribution sweeping: Given N rectangles in the plane, compute the measure (area) of their union.