CPS 238:
I/O-Efficient Algorithms

Homework/Take-home exam

Due Thursday 12/6 in class

Merging m-way

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.)

Search trees

1. Show how to rebalance (a, b)-trees ($b \geq 2a$) top-down, that is, how to do the rebalancing while searching down the tree for the leaf where an update should be performed.

2. Show how to use the top-down balanced (a, b)-tree as a buffer tree.

Graph algorithms

1. Design an $O(sort(V))$ algorithm for computing the number of descendents of each vertex in a tree.

2. Show that $\Omega(perm(V)$ is a lower bound on the problem in the "indivisibility model" when we only consider algorithms that for each edge (v, w) in the tree at some point holds the original vertices v and w in main memory at the same time.

Distribution sweeping

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