CSCI 5020 External Data Structures: Assignment 1
(Submission Deadline: 30 June 2011)

In the following problems, $B$ equals the block size; and you can assume that the amount of memory $M$ is at least $B^2$ if necessary.

**Problem 1 (25%).** Let $\mathcal{I}$ be a set of $N$ intervals in $\mathbb{R}$. Given a interval $q$ in $\mathbb{R}$, a range query reports all the intervals in $\mathcal{I}$ intersecting $q$ (two intervals intersect if they cover at least one common value). Describe how to maintain $\mathcal{I}$ in a fully dynamic structure that consumes $O(N/B)$ space, answers a query in $O(\log_B N \cdot \log_B N + K/B)$ I/Os (where $K$ is the number of intervals reported), and supports each insertion and deletion in $O(\log_B N)$ I/Os.

**Problem 2 (35%).** Let $\mathcal{I}$ be a set of $N$ intervals in $\mathbb{R}$ that do not partially intersect each other. That is, for any two intervals $s_1, s_2$ in $\mathcal{I}$, exactly one of the following can happen:

- they are disjoint;
- $s_1$ is covered by $s_2$ (i.e., $s_1 \subseteq s_2$);
- $s_2$ is covered by $s_1$.

Given a value in $\mathbb{R}$, an order-sensitive stabbing query reports all the intervals of $\mathcal{I}$ covering $q$ in ascending order of their left endpoints. Describe a fully dynamic structure to index $\mathcal{I}$ such that the structure consumes $O(N/B)$ space, answers a query in $O(\log_B N \cdot \log_B N + K/B)$ I/Os (where $K$ is the number of intervals reported), and supports each insertion and deletion in $O(\log_B N)$ I/Os.

**Problem 3 (30%).** Let $S$ be a set of $N$ horizontal segments in $\mathbb{R}^2$. Given a point $q$, a ray shooting query reports the first segment in $S$ above $q$, namely, the first segment of $S$ hit by a vertical ray emanating upwards from $q$. Describe a fully dynamic structure that consumes $O(N/B)$ space, answers a query in $O(\log_B^2 N)$ I/Os, and supports the insertion and deletion of a segment in $S$ using $O(\log_B^2 N)$ amortized I/Os.

**Problem 4 (10%).** Improve the update time of your structure in Problem 3 to $O(\log_B N)$ amortized, while still ensuring the same space and query performance.