CSCI 5020 External Data Structures: Exercise List 3

In the following problems, $B$ is the block size, and $M$ is the memory capacity.

**Problem 1 (Bundled predecessors).** Let $S_1, ..., S_{\sqrt{B}}$ be $\sqrt{B}$ sets of integers. Given a value $q$, a query reports the predecessor of $q$ in every $S_i$ ($1 \leq i \leq \sqrt{B}$), namely, $\sqrt{B}$ predecessors need to be output. Describe a structure that consumes $O(N/B)$ space, and answers a query in $O(\log_B N)$ I/Os, where $N = \sum_i |S_i|$. 

**Problem 2.** In Lecture 5, we discussed the construction of the persistent B-tree. Describe how to maintain a structure so that, given a timestamp $t$, we can find in $O(\log_B N)$ I/Os the root of the B-tree at time $t$ in the persistent B-tree, where $N$ is the number of updates performed on the persistent B-tree. Your structure must consume $O(N/B)$ space.

**Problem 3 (Tree packing in external memory).** You are given a tree with $N$ nodes (a tree is defined as a connected, undirected, graph without any cycle). Let $r$ be the root of the tree. Given any node $u$ in the tree, there is a unique path from $u$ to $r$, which we denote as $u \sim r$. Pre-process the tree into a structure of size $O(N/B)$ such that, given any node $u$, we can report $u \sim r$ in $O(\log_B N + K/B)$ I/Os, where $K$ is the number of nodes on $u \sim r$.

**Problem 4*.** Improve your solution to Problem 3 so that a query can be answered in $O(1 + K/B)$ I/Os, whereas the size of your structure remains linear.