CSCI 5020 External Data Structures: Exercise Spoilers

1 Exercise 1

Problem 1. Do external sort except that, in merging, combine tuples with identical keys. Argue that the cost is as desired.

Problem 2. Use distribution sort with \( \sqrt{m} \) slabs. Use linked lists to manage rectangles.

2 Exercise 2

Problem 1. Hint as given in the exercise description.

Problem 2. Adapt the following proof for the statement that (in internal memory) \( \Omega(\log N) \) comparisons are needed to find the predecessor of \( q \) in a set \( S \) of \( N \) elements. Let \( P \) be the set of elements that can possibly be the result. At the beginning, \( |P| = N \). After a comparison, \( P \) can be divided into \( P_1 \) and \( P_2 \) such that \( P_1 \cup P_2 = P \). Hence, either \( P_1 \) or \( P_2 \) has a size of \( |P|/2 \).

3 Exercise 3

Problem 1. Use a B-tree with internal fanout \( \Theta(\sqrt{B}) \).

Problem 2. Pay attention to the version copies of the roots.

Problem 3. Try to give each node of the tree an interval such that a node \( u_1 \) is an ancestor of another node \( u_2 \) if and only if the interval of \( u_1 \) contains that of \( u_2 \).

Problem 4. Use a persistent B-tree. Associate each node with a pointer to somewhere in the tree that allows you to start churning out the result immediately.

4 Exercise 4

Problem 1. Use internal fanout \( \Theta(B^{0.25}) \).

Problem 2. First build the base tree and then the secondary structures top-down.

Problem 3. Use a B-tree to index all the data rays by their y-coordinates, and store additional information at each routing element.

5 Exercise 5

Problem 1. Use filtering search.

Problem 2. Integrate your solution to Problem 1 and the external priority search tree.

Problem 3. Combine your solution to Problem 2 and the external priority search tree.