CS 240E: Structures and Data Management **Winter 2021**

Tutorial 7: More Hashing, Quad-trees

1. Design a dictionary data structure to store key-value-pairs with uniformly distributed integer keys such that the operations for search, insert, and delete have worst case $O(\log n)$ runtime and $O(1)$ expected runtime (ignoring rehashing).

2. Assume that $p_1, \ldots, p_n$ are n distinct points in 2D, and that the coordinates of each p_i are 32-bit 01-strings. Give an algorithm that builds a quad-tree of these points in $O(n)$ time.

3. Build a quadtree using the following points: (1, 4), (2, 5), (3, 2), (4, 7), (7, 3), (6, 1), (5, 6), (3, 7).

4. Let S be a quadtree consisting of 2d points. Given a radius r and a center p, method *ballRangeQuery*(S, p, r) returns all points in S with Euclidean distance from point p less than or equal to r. Describe how to implement method *ballRangeQuery*(S, p, r) with a quadtree. Explain the idea and the worst case complexity of your algorithm. You can assume you have a $O(1)$ method that tells you whether a query cirle A and a rectangular region B intersect.