York University EECS 4101 August 4, 2023

# Homework Assignment #10
# Due: August 14, 2023 at 10:00 p.m.

[4] **1.** You are given a collection of $n$ data points $(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)$ in a 3-dimensional space and you wish to cluster them. That is, you want to divide the points up into disjoint clusters as follows. Two points *must* be put in the same cluster if the distance between them is less than or equal to $T$. You want as many clusters as possible, subject to the preceding rule.

Give an algorithm that performs this clustering. Give upper bounds on the time and space required by your algorithm.

**2.** Suppose we use trees to implement the union-find ADT. We use path compression. But instead of using union by rank, we use union by size: each root has a *size* field that stores the size of tree rooted at that node, and when a UNION operation connects two roots, the one with the smaller *size* is made the child of the other. (Ties can be broken arbitrarily.)

Once a node becomes a child of another node, we do not bother to update its *size* field ever again because (a) we never use the *size* field of that node ever again, and (b) it would be too hard to figure out how the size of the node's subtree changes when we do path compression.

[2] **(a)** Show that the worst-case time for an individual FINDSET or UNION operation is $O(\log n)$, where $n$ is the number of nodes in the data structure.

[5] **(b)** We want to argue that the amortized $O(\log^* n)$ time bound shown in class still holds if we do path compression and union by size. Consider a sequence of $m$ MAKESET, FINDSET and UNION operations, starting from an empty data structure. Let $n$ be the number of MAKESET operations in the sequence.

Define the *grade* of a node $v$ to be the base-2 logarithm of the value that $v.size$ has *at the end of the entire sequence of operations.* (The algorithm does not compute the grade of any node; the grade of the node is just something we define for the sake of doing our analysis.) Explain why the grades of nodes satisfy the same properties as the ranks would satisfy (if we were doing union by rank), namely:

(i) If a node $v$ is the child of a node $u$ at any time, then $v.grade \leq u.grade - 1$.

(ii) If path compression changes a node $v$'s parent from node $u$ to node $u'$, then $u.grade \leq u'.grade - 1$.

(iii) For all natural numbers $r$, at most $\frac{n}{2^r}$ nodes have grades in the interval $[r, r+1)$.

These properties have been reworded slightly because grades might not be integers, whereas ranks always are integers.

Hence, if we divide nodes into blocks according to the value of $\log^*(grade)$, and divide up the steps of each path traversal during a FINDSET or UNION into block steps and ordinary steps (as described in class), the proof we did in class would then show that the total time for the sequence of $m$ operations is $O(m \log^* n)$. (Think through the proof to doublecheck this, but you don't have to write anything down about this.)