York University CSE 4101/5101 April 14, 2009

Homework Assignment #5
Due: April 23, 4:00 p.m.

1. In class, we discussed a union-find data structure that uses a tree to represent each disjoint set, and uses union by rank and path compression. We showed that the total time to perform a sequence of m Find-Set, Make-Set and Union operations (of which n are Make-Sets) is $O(m \log^* n)$.

Now suppose that, instead of doing path-compression during Find-Set operations, we do path-grandparenting: During a Find-Set(x), as we traverse the path of nodes from x to the root, we update the parent pointer of each node along the path to point to its grandparent. The pseudocode for Find-Set with path-grandparenting appears below.

```
Find-Set($x$)
    loop until $p[x] = x$ % $p[x] = x$ means $x$ is a tree root
        $p[x] \leftarrow p[p[x]]$
        $x \leftarrow p[x]$
    end loop
    return $x$
end Find-Set
```

(We are using the same Union and Make-Set procedures as are shown on page 508.)

Give a good upper bound on the total time to perform a sequence of m Find-Set, Make-Set and Union operations (of which n are Make-Sets), starting from an empty data structure. Prove your answer is correct.