**CS146 HW1**

**Problem 1:**

Ans: As you can see in the union() in

public void union(int p, int q) {

if (connected(p, q)) return;

for (int i = 0; i < id.length; i++)

if (id[i] == id[p]) id[i] = id[q];

count--;

}

The value of id[p] changes to id[q] and if a > b with id[a] = id[b] will not be updated equal to id[q].

Examples: union(0,1) => 1 1 2 3 4 5 …

Union(0,2) => 2 1 2 3 4 5…

**Problem 2:**

Ans: If we set id[root(p)] = q instead of id[root(q)] the resulting algorithm would be correct. If we do this we increase the tree height. No there are no performance consequences.

**Problem 3:**

Ans: Coded (Program works for all inputs)

**Problem 4:**

Ans:

**Quick Union**

|  |  |  |
| --- | --- | --- |
| **smallUF.txt took : 0.154ms** | **mediumUF.txt took: 0.241ms** | **largeUF.txt took a lot of time to compute** |
|  |  |  |
|  |  |  |

**QuickUnionPathCompressionUF**

**smallUF.txt took: 0.136ms**

**mediumUF.txt took: 0.24ms**

**largeUF.txt took a lot of time to compute.**

Quick Union will have O(nm) in worst case and Quick Union Path compression will have O(n+mlogn).

**Problem 5: Coded**

**Problem 6: Coded**

**Problem 7: Coded**