CS 561, HW5

Prof. Jared Saia, University of New Mexico

Due: November 5th

1. Suppose we are maintaining a data structure under a series of operations. Let $f(n)$ denote the actual running time of the nth operation. For each of the following functions f, determine the resulting amortized cost of a single operation *using the potential function method. Make sure you give your potential function, show that it is valid, and calculate the amortized costs in all cases.*

   (a) $f(n) = n$ if n is a power of 2, and $f(n) = 1$ otherwise.

   (b) $f(n) = n^2$ if n is a power of 2, and $f(n) = 1$ otherwise.

   (c) $f(n)$ is the largest power of 2 that divides n.

2. Problem 17-2 (Making Binary Search Dynamic)

3. Professor Curly conjectures that if we do union by rank, *without path compression*, the amortized cost of all operations is $o(\log n)$. Prove him wrong by showing that if we do union by rank without path compression, there can be m MAKESET, UNION and FINDSET operations, n of which are MAKESET operations, where the total cost of all operations is $\Theta(m \log n)$.

4. Problem 22-4 (Reachability) [1]

5. Professor Moe conjectures that for any connected graph G, the set of edges $\{$(u,v) : there exists a cut (S,V-S) such that (u,v) is a light edge crossing (S, V-S)$\}$ always forms a minimum spanning tree. Given a simple example of a connected graph that proves him wrong.

6. Exercise 23.1-2 ("Professor Sabatier conjectures")

[1] The answer to this problem can be used in an efficient randomized algorithm for estimating the *number* of vertices that are reachable.

7. Exercise 23.1-3 ("Show that if an edge (u,v) is contained in some minimum spanning tree")