Mar 31

(1) The following operations manipulate piles of “rocks”. All rocks are identical, except that each rock has a “grade”, which is an integer from 1 to $n$.

- CreateRock creates a new rock of grade 1.
- Promote($g$) changes one rock from grade $g$ to grade $g + 1$.
- Merge($g, h$) removes one rock of grade $g$ and one rock of grade $h$, and creates a rock of grade $g + h$.
- Delete($g$) removes a rock of grade $g$.
- $x = \text{MaxGrade}()$ returns the highest grade of any rock. It does not change any rocks or grades.

Implement this abstract datatype using only an integer Max and an array Rocks[$1 \ldots n$], where Rocks[$g$] is the number of rocks of grade $g$. You may assume that Rocks is initially all zero, and that no operations that would cause errors are ever done. You are to show how to implement these operations so that all except Delete run in constant real time, and give a potential function proof that they all run in constant amortized time.

(2) Modify the Bellman Ford shortest path algorithm to find a negative cycle in a directed graph. In other words, if a negative cycle exists then how would you identify it?

(3) Consider a directed graph with edges labelled with a non-negative real number. How do we find a cycle such that the product of the edges is $< 1$.

(4) Problem 6 (page 191) Chapter 4.

(5) You are given $n$ numbers $x_1, \ldots, x_n$ (these are stored in an unsorted vector of length $n$). The goal is to find the smallest element and also the second smallest element.

(a) Prove that if $n$ is a power of 2 then you can manage with $n + \log n - 2$ comparisons.
(b) How many comparisons do you need if $n$ is not a power of 2 (but an arbitrary positive integer)?