Algorithms
Homework 6

Saad Mneimneh
Computer Science
Hunter College of CUNY

Readings
Based on lectures for augmenting red-black trees, dynamic programming and greedy algorithms.

Problem 1
Describe how you will augment a redblack tree to support the operation even_successor(x), which finds the node y in the tree with the smallest key k such that $k > x.key$ and k is even. We assume all keys are distinct. The even successor operation should run in $O(\log n)$ time, where n is the number of keys stored in the tree. Describe the operation clearly in English and support it with pseudocode.

Problem 2
Consider the problem of partitioning an integer m into a sequence of k integers, each in $\{1, \ldots, n\}$. For instance, here are the partitions of 8 into 3 parts, each in $\{1, 2, 3, 4\}$.

$$1, 3, 4$$
$$1, 4, 3$$
$$2, 2, 4$$
$$2, 3, 3$$
$$2, 4, 2$$
$$3, 1, 4$$
$$3, 2, 3$$
$$3, 3, 2$$

$$3, 4, 1$$

$$4, 1, 3$$

$$4, 2, 2$$

$$4, 3, 1$$

Let $p(m, k)$ be the number of partitions given m, k, and n.

(a) Find a recursive definition of $p(m, k)$

(b) Compute $p(m, k)$ by implementing a recursive function that mimics the recursive definition.

(c) Compute $p(m, k)$ using memoization. You will need a two dimensional array to store the values of $p(m, k)$.

Problem 3

There are n people teaming up for a hiking trip. Each person can carry two bags, so there are $2n$ bags in total. Assume bag i has weight b_i. There are many reasonable (or unreasonable) ways one could optimize the assignment of bags to people. Parts (a) and (b) explore two alternatives, the two questions are independent of each other.

(a) Describe an algorithm to assign each person two bags in such a way to minimize the heaviest load. In other words, if person k is carrying bag i and bag j, let $w_k = b_i + b_j$ be the sum of the two weights. We want to keep the largest w at a minimum.

(b) Every person will carry one bag in each hand. We want both hands to be well balanced, so if person k is carrying bag i and bag j, let $d_k = |b_i - b_j|$ be the difference between the two weights. Describe an algorithm that minimizes the largest difference, i.e. that keeps the largest d at a minimum.

Problem 4

Professor Saad is old fashioned and likes to use chalk during class (although he is forced to use the ipad). He has a bag of chalk, initially containing n chalks. He needs half a chalk per class. Therefore, on each day, he blindly grabs a chalk from the bag. If he pulls a whole chalk, he breaks it in two halves, returns one half to the bag, and uses one half. If he grabs a half chalk, he uses that one. After $2(n-1)$ days, his bag will contain either one whole chalk, or two halves of a chalk.

(a) What is the probability of ending with a whole chalk? Define $p(i, j)$ as the probability of ending with a whole chalk given that he starts with i wholes and j halves. Obviously, $p(0, j) = 0$. Define a recurrence for $p(i, j)$

and suggest a dynamic programming algorithm to compute $p(n, 0)$.

(b) Saad will either break a chalk or not on each day. This defines a binary sequence. What is the number of possible binary sequences that can be produced? Define $b(i, j)$ as this number when he starts with i wholes and j halves. The quantity $b(i, j)$ will have a similar recurrence to $p(i, j)$.