Algorithms, Spring 1997

Assignment 3

Due in two weeks.

5.17, 5.19, 6.53 of Manber

Problem 4

A gambler has a units of money at time 0 and gambles for n time units. At the i th time, the gambler's assets increase by 1 or decrease by 1; the probability of increse is p_i and decrease is $1 - p_i$ (independent of what happened before.) Given $a, n, p_1, p_2, \ldots p_n$, describe an $O(n^2)$ time algorithm to compute the probability that at time n, the gambler's assets are at least a.

Problem 5

For this problem, a text is a sequence of words. We are given the lengths $l_1, l_2, \ldots l_n$ of n words that occur (in that order) in a text. Our task is to arrnage them into lines with one or more spaces between two successive words. The length of each line is not to exceed a given bound b . We want a pleasing arrangment. The first requirement (for a pleasing arrangement) is that each word occurs wholly on a single line. [Assume $l_i \leq b$ for all i.] The second requirement is that we must minimize the total penalty of all the lines. The penalty of a line is defined as follows : if a line contains words $i, i+1, \ldots j$, then the penalty on the line is sqaure of the number of "extra" blank spaces put into the line , namely $(b - j + i - l_i - l_{i+1} - \ldots l_j)^2$. [The exact penalty is not imporant; just think of it as some function of the number of extra blank spaces.] Design a $O(nb)$ time algorithm to find a pleasing arrangement. [Assume that each arithmetic operation - addition, subtraction, multiplication and division can each be done in time $O(1)$.]

Problem 6 Suppose we have a square $n \times n$ grid as shown below. Each edge has associated with it a weight which is given to us. A "path" is defined as a sequence of edges starting at the bottom left corner and ending at the top right corner with each edge going right (east) or up (north) from the end point of the previous edge. (An example path is shown below.) So, there are exactly $\binom{2n}{n}$ paths. The weight of a path is the sum of the weights of the edges in it. Describe an $O(n^2)$ time algorithm to find the average weight of a path. (The total weight of all paths divided by the number of paths.) [Assume that each arithmetic operation - addition, subtraction, multiplication and division can each be done in time $O(1)$.]