Dice Probabilities

We wish to find the probability that rolling k 6-sided fair dice will result in a sum S. Devise an algorithm to find this probability.

Knight Moves

Given an 8×8 chessboard and a knight that starts at position $a1$, devise an algorithm that returns how many ways the knight can end up at position xy after k moves. Knights move ± 1 squares in one direction and ± 2 squares in the other direction. In other words, knights move in a pattern similar to a "L".

Note: on a chessboard, rows are labeled from 1-8 and columns are labeled from $a - h$.

Encoding

Suppose we encode lowercase letters into a numeric string as follows: we encode a as 1, b as 2, ..., and z as 26. Given a numeric string S of length n, develop an $O(n)$ algorithm to find how many letter strings this can correspond to. For example, for the numeric string 123, the algorithm should output 3 because the letter strings that map to this numeric string are abc, lc, and aw.

Exact Acorn Dropping

We have a acorns and b branches. We want to compute the *exact* minimum number of drops needed (in the worst case) to find the highest branch in which an acorn will not break after dropping. Assume that all of the acorns are the same strength; if one acorns breaks after dropping from a branch, all acorns will break after dropping from that branch. Design an algorithm that returns the minimum number of drops needed to accomplish this task in the worst case, without actually dropping any acorns.

Rod Cutting

Suppose we have a rod of length k, where k is a positive integer. We would like to cut the rod into integer-length segments such that we maximize the *product* of the resulting segments' lengths. Multiple cuts may be made. For example, if $k = 8$, the maximum product is 18 from cutting the rod into three pieces of length 3, 3, and 2. Write an algorithm to determine the maximum product for a rod of length k.