1. **Binomial Coefficients:** Give an efficent algorithm to compute $\binom{n}{k}$, the binomial coefficient of n and k. We define $\binom{n}{k}$ to be the number of ways to choose k items from a set of n distinct items. What is the running time of your algorithm?

2. **Making Change:** We have a currency system that has n coins with values $v_1, v_2, \ldots v_n$, where $v_1 = 1$ (so we can make change for any amount). We want to make change for the amount Y, in such a way that the total number of coins used is minimized. Note that the greedy algorithm for this does not always work; as discussed in a previous HBS, we can come up with a set of coin values such that the greedy algorithm does not always give the optimal solution.

   More formally, give an algorithm that minimizes the quantity $\sum_{i=1}^{n} x_i$ subject to the constraint that $\sum_{i=1}^{n} x_i v_i = Y$.

3. **Catalan Paths:** Consider the number of paths on a 2-dimensional grid from $(0, 0)$ to (n, n), where every step of the path moves one square right, or one square up. (That is, from (a, b), you can only move to $(a + 1, b)$ or $(a, b + 1)$.) It is not too hard to see that there are $\binom{2n}{n}$ ways to go from the origin to (n, n).

   Now consider only paths that never go *above* the diagonal $y = x$. That is, we want to find the number of ways to travel from $(0, 0)$ to (n, n) such that no intermediate point is of the form (a, b), where $b > a$. A path which never rises above the diagonal is called a *Catalan* path; after the Belgian mathematician Eugene Charles Catalan. Find an dynamic-programming algorithm that computes the number of Catalan paths from the origin to (n, n). (This is called the n^{th} *Catalan number*; the Catalan numbers have lots of nice properties, and show up in all kinds of interesting and unexpected places.)