Jacob Alspaw
EECS 340
Assignment #6

15.1-2

The density of a rod of length i to be p_i=i, that is, its value per inch. The greedy strategy for a rod of length n cuts off a first piece of length i, where 1 ≤ i ≤ n, having maximum density. It then continues by applying the greedy strategy to the remaining piece of length n - i.

Counter example:

Length (n)	1	2	3	4
Price (p_i)	1	6	10	8
Density (p_i / i)	1	3	10/3	2

In the above example, the length of the rod is four. When applying the greedy strategy to the rod, a rod length of 3 will first be chosen because it has the highest price density. The remaining piece will be of length 1 which means the total price obtained from the rod is $p_3 + p_1$ = 10 + 1 = 11 units. But this selection is not optimal! The optimal solution is 2 pieces of length 2 for a price of $p_2 + p_2$ = 6 + 6 = 12 units. The greedy algorithm would have missed the optimal solution because price density of length 2 is not the greatest.

15.1-3

```
CUT-ROD(p, n) {  //where p is an array of prices for rods of length n
        if n = 0, then
                return 0
        q ← -∞   //where q is the current revenue after cutting and selling the rod
        for j←1 to n do
                q←max(q, p[ j ] – c[ j ] + CUT-ROD(p, n – j)) //where c[j] is the cost to cut piece j
        return q   //where q is the value of the optimal solution
```

In the above recursive call, there are many sub-problems which are solved again and again. Since the same sub-problems are called again, this problem has the overlapping sub-problems property. Also, the “max” function will find the optimal solution to the rod cutting problem. So the Rod Cutting problem has both properties of a dynamic programming problem.

15.3-6

We have n different currencies, numbered 1, 2, . . . , n, where you start with currency 1 and wish to wind up with currency n. You are given, for each pair of currencies i and j, an exchange rate r_{ij}, meaning that if you start with d units of currency i, you can trade for dr_{ij} units of currency j. Let c_k be the commission that you are charged when you make k trades.

If $c_k=0$ for all k = 1,2, . . . , n:
Consider the starting currency 1 and the ending currency *n*. There may exist a currency *k* in the exchanges from 1 to *n* such that $dr_{1k}r_{kn} > dr_{1n}$. The trade would go from 1 to *k* to *n* where *k* can have *n*-2 choices.

The currency problem further divides into two sub-problems of trading from currency 1 to currency *k*. There may exist a currency *i* such that $dr_{1i}r_{ik} > dr_{1k}$. The trade would go from 1 to *i* to *k* where *i* can have n-3 choices. This sub-problem could divide into even further sub-problems of currency 1 to currency *i* and currency *i* to currency *k*.

The remaining sub-problem is trading from currency *k* to currency *n*. There may exist a currency *j* such that $dr_{kj}r_{jn} > dr_{kn}$. The trade would go from *k* to *j* to *n* where *j* can have *n*-3 choices. This sub-problem could divide into even further sub-problems of currency *k* to currency *j* and currency *j* to currency *n*.

There is an optimal solution to each sub-problem such that the conversion is maximized. The solution to these sub-problems' parents are the combined maximized solutions produced by the sub-problems themselves. Upon combining these sub-problems, the main problem becomes optimized. This means that the solution exhibits optimal substructure.

If $c_k \neq 0$ for all k = 1,2, . . . , n:
Consider the starting currency 1 and the ending currency *n*. There may exist a currency *k* in the exchanges from 1 to *n* such that $dr_{1k}r_{kn} > dr_{1n}$. The trade would go from 1 to *k* to *n* where *k* can have *n*-2 choices. Even though the conversion rate from 1 to *k* to *n* would produce a higher conversion rate than directly from currency 1 to *n*, the solution may not be optimal because the commission rate c_k is an arbitrary value. There is no guarantee that the sub problems will produce optimal solutions because the commission rates are random values. Because there is no guarantee for an optimal solution in the sub-problems, then there is no guarantee for an optimal solution in the main problem, so the exchanges would not exhibit the optimal substructure property.