CS 561, HW5

Prof. Jared Saia, University of New Mexico

1. A frog is jumping across a line of lily pads. It starts at lily pad 1. When the frog is at lily pad i for any $i \geq 1$, it jumps to lily pad $i+1$ with probability $1/2$ and to lily pad $i+2$ with probability $1/2$.

   (a) Let $p(i)$ be the probability that the frog ever visits lily pad i, for any $i \geq 1$. Write a recurrence relation for $p(i)$. Don't forget the base case(s).

   (b) Use annihilators to solve for a general solution to your recurrence relation.

   (c) Use the base case(s) of your recurrence to solve for an exact solution.

   (d) Now, let X be a random variable giving the number of lily pads between lily pad 1 and n that the frog visits, for some number n. Compute $E(X)$ by using: linearity of expectation, indicator random variables, and your solution to the recurrence $p(i)$ that you found above.

2. As in the last homework, a thief repeatedly robs the same bank. To avoid capture, he never robs the bank fewer than 10 days after the last robbery. He has obtained information, for the next n days, on the amount of money b_i that is held at the bank on day i.

   But now, the thief is lazy: (1) for each day, there is an integer value giving the amount of work w_i that the thief must perform to rob the bank on that day (due to the amount of security on that day); and (2) there is an additional constraint that the sum of work the thief ever performs is less than some value W.

   Let $r(i,j)$ be the maximum amount of revenue obtainable on days 1 through i, with at most j total work.

   (a) Give a recurrence relation for $r(i,j)$.

(b) Describe a dynamic program based on this recurrence. What is the runtime of your algorithm?

3. You have n feet of cable to be cut it into pieces for resale. On a given day, pieces of length 1, 3, and 7 can resell for values of v_1, v_2 and v_3. You want to cut the cable into pieces with maximum total resell value. For example, if $n = 14$, and $v_1 = 1$, $v_2 = 4$ and $v_3 = 8$, then the optimal cutting is: 4 pieces of length 3, and 2 pieces of length 1, for a total resell value of $4 * 4 + 2 * 1 = 18$.

   You decide to solve this problem with dynamic programming. For any number $i \in [0, n]$, let $m(i)$ be the maximum resell value you can get from optimally cutting a cable of length i. Write a recurrence relation for $m(i)$. Don't forget the base case(s)

4. Now consider a variant of the above problem where the maximum number of cuts you can make is some integer k. As before, pieces of length 1, 3, and 7 resell for values of v_1, v_2 and v_3. Any pieces of other lengths have zero value. For example, if $n = 14$, $k = 1$ and $v_1 = 1$, $v_2 = 4$ and $v_3 = 8$, then the optimal cutting is: 2 pieces of length 7 for a total resell value of $2 * 8 = 16$.

   Write a recurrence relation for a dynamic program for this variant. In particular, for any numbers $i \in [0, n]$ and $j \in [0, k]$, let $m(i, j)$ be the maximum resell value you can get from cutting a cable of length i with at most j cuts. Write a recurrence relation for $m(i, j)$. Don't forget the base case(s).

5. **Chocolate with Friends.** You have a chocolate bar consisting of n chunks aligned in a single row. Each chunk, i, for $1 \leq i \leq n$ has some positive value v_i (for example, chunks with high nougat content are more valuable than those without!)

   You must break the bar into exactly k *parts* to share with your friends, where each part consists of some number of contiguous, unbroken chunks. The value of a part is the sum of the value of all chunks in that part. Your (greedy) friends chose their parts first, and you get the part remaining, i.e the part of smallest value. Thus, your goal is to break the bar in such a way that you maximize the value of the minimum value part.

   Write a recurrence relation and give a dynamic program to solve this problem. What is the runtime of your algorithm?