Recitation 10: Dynamic Programming

1 Avoiding Substrings

Suppose we have the standard 26-letter English alphabet, $\Sigma = \{a, b, \ldots, y, z\}$. Let W_n be the set of strings of length n which do not contain the word "yay":

$$W_n = \{\omega \in \Sigma^n : \omega_i \omega_{i+1} \omega_{i+2} \neq \text{yay},\ \forall i = 1, \ldots, n-2\}.$$

Write a recurrence for $f_n = |W_n|$, including base cases, to count the number of character strings of length n that do not contain the word "yay".

(The notation Σ^n means "the set of any n characters from the alphabet Σ concatenated". So $\{x, y\}^3 = \{xxx, xxy, xyx, xyy, yxx, yxy, yyx, yyy\}$.)

2 Trains

You've decided to leave CS to pursue a career in train robbery (it's the next big thing!). You've been observing the train schedules in the Boulder area, and have a pretty good idea of what trains will be running in the next month, and the approximate value of each train's cargo.

Over the next month, you know there will be n trains running in your target area, with train i carrying cargo worth some value v_i. Unfortunately, you expect the law to be close on your heels; you've decided after each heist it's best to lay low and leave the next 2 trains alone to avoid getting caught.

Give a dynamic programming algorithm to determine the maximum amount of loot you'll be able to make off with in the next month.

a. Identify the subproblem to solve.

b. Define a recurrence for V_i, the total value of loot you can boost over trains $i, i+1, \ldots, n$. Include your base cases.

c. Say there are 12 trains running this month, with values

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
20	18	6	8	15	8	4	23	7	9	13	16

Use your recurrence to compute the maximum loot value you can get this month. What is the maximum value? How could you modify this to give a schedule for your train robbery, as well as your optimal value?