CSCI 3104: Algorithms
Spring 2022
Recitation #9 - Dynamic Programming I

Problem 1

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 $\{x, y\}^3 = \{xxx, xxy, xyx, xyy, yxx, yxy, yyx, yyy\}$.)

Problem 2

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?

Please don't rob trains.

Problem 1

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 $\{x, y\}^3 = \{xxx, xxy, xyx, xyy, yxx, yxy, yyx, yyy\}$.)

$W_0 = 0$

$W_1 = 26 \Rightarrow a \ldots z$

$W_2 = 26^2 \Rightarrow a \ldots z$, a string from W_1 $\Rightarrow$ there're 26 strings.

$W_3 = 25 W_2 + 25 W_1 + 25$

$\neq y$ $\{a \ldots z$, $\boxed{\text{a string from } W_2}$

y, $\neq a$ $\{b \ldots z$ $\boxed{\text{a string from } W_1}$

a, $\neq y$ $\{a \ldots z$ $\cancel{y}$

$W_4 = 25 W_3 + 25 W_2 + 25 W_1$

$\neq y$ $\{a \ldots z$, $\boxed{\text{a string from } W_3}$

y, $\neq a$ $\{b \ldots z$ $\boxed{\text{a string from } W_2}$

a, $\neq y$ $\{a \ldots z$ $\boxed{\text{a string from } W_1}$ $\cancel{y}$

$\neq y$ $\{a \ldots z$ $\boxed{W_{n-1}}$

y $\neq a$ $\{b \ldots z$ $\boxed{W_{n-2}}$

a $\neq y$ $\{a \ldots z$ $\boxed{W_{n-3}}$ $\cancel{y}$

$$f_n = \begin{cases} 0, & n \leq 0 \\ 26, & n = 1 \\ 576, & n = 2 \\ 25 \cdot f_2 + 25 f_1 + 25, & n = 3 \\ 25 \cdot f_{n-1} + 25 f_{n-2} + 25 f_{n-3}, & n \geq 4 \end{cases}$$

Problem 2

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.

when the ith train comes,
we need to decide either 1) rob it.
2) let it go by.

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.

$$X_i = \begin{cases} \max(X_{i+1},\ v_i + X_{i+3}), & 1 \le i \le n \\ 0, & i > n \end{cases}$$

where $X_i = V_i$

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?

Please don't rob trains.

$X_{15} = 0$

$X_{14} = 0$

$X_{13} = 0$

V_{10}	V_{11}	V_{12}
9	13	16

$\Rightarrow$

$X_{12} = \max(X_{13},\ V_{12} + X_{15}) = 16$

$X_{11} = \max(X_{12},\ V_{11} + X_{14}) = 16$

$X_{10} = \max(X_{11},\ V_{10} + X_{13}) = 16$

V_7	V_8	V_9	X_{10}	X_{11}	X_{12}
4	23	7	16	16	16

$\Rightarrow$

$X_9 = \max(X_{10},\ V_9 + X_{12}) = 23$

$X_8 = \max(X_9,\ V_8 + X_{11}) = 39$

$X_7 = \max(X_8,\ V_7 + X_{10}) = 39$

V_4	V_5	V_6	X_7	X_8	X_9
8	15	8	39	39	23

$\Rightarrow$

$X_6 = \max(X_7,\ V_6 + X_9) = 39$

$X_5 = 54$

$X_4 = 54$

V_1	V_2	V_3	X_4	X_5	X_6
20	18	6	54	54	39

$\Rightarrow$

$X_3 = 54$

$X_2 = 72$

$X_1 = 74.$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
	74	72	54	54	54	39	39	39	23	16	16	16
Rob(i).	Y	Y	N	N	Y	N	N	Y	Y	N	N	Y

With this table, we can find an ordering of heists to commit by starting at $i=1$. to n, where $n=12$.

if rob(i) = Y, we add train i to our to-rob list and skip to $i+3$.

rob(i) = N, we move to $i+1$.

This gives us the robbery schedule of trains 1, 5, 8, 12.

$\Rightarrow$ The optimal value is 74.