Recitation 11: Dynamic Programming

1 Bamboozled

We say that the set $B \subset \{1, 2, 3, ..., n\}$ is ***bamboozled*** if for every three consecutive numbers $i, i+1, i+2$ (for $1 \leq i \leq n-2$) either one or two of the numbers are in B.

For example, the set $\{1, 2, 4, 5\} \subset \{1, 2, 3, 4, 5, 6\}$ is bamboozled, but the set $\{2, 3, 4, 6\} \subset \{1, 2, 3, 4, 5, 6\}$ is not.

Problem 1.1. Given a number n, count the number of bamboozled subsets of $\{1, 2, ..., n\}$

a. Let $T(n)$ be the number of bamboozled subsets of $\{1, 2, ..., n\}$. Give a recurrence for $T(n)$.

Problem 1.2. Given a number n, and a list of weights $w_1, w_2, ..., w_n$, find a bamboozled subset $B \subseteq \{1, 2, ..., n\}$ which maximizes the total weight

$$\sum_{i \in B} w_i$$

a. Identify specific subproblems

b. Identify the relationship between subproblems, and use this to write down a recurrence relation. What are the base cases?

c. Fill out a lookup table with the optimal values to these subproblems

d. Use backtracking to find an optimal bamboozled set B

e. Analyze the runtime of your lookup table algorithm in terms of n. How does this compare to the runtime of brute force?

It may help to do this with a concrete example before writing down an abstract algorithm. Try it with the set $\{1, 2, 3, 4, 5, 6, 7\}$ and weight vector

$$w[1..7] = [4, 6, 2, 8, 4, 3, 6]$$

2 Aliens

CU's top-secret alien communications program has hired you to decode some radio signals they recently picked up. It turns out two different groups of aliens have been sending us messages for YEARS, and we haven't noticed! You've figured out the problem: the two species' transmissions have gotten mixed up.

Specifically, alien species A has been transmitting the signal $a_1a_2\dots a_p$, while species B has been transmitting the signal $b_1b_2\dots b_q$. Each species has been transmitting their message non-stop, but the signals have gotten jumbled: instead of receiving $a_1\dots a_pa_1\dots a_pa_1\dots a_p\dots$ and $b_1\dots b_qb_1\dots b_qb_1\dots b_q\dots$, we've been receiving a mix of the two strings' bits, which might look something like: $a_1a_2b_1b_2a_3a_4a_5b_3a_6b_4\dots$. The bits are all there, and they're in order - the two strings are just mixed up.

You have an idea what the strings are, but you want to write a program to confirm that you've gotten it right. Write a dynamic program that takes in a received signal $c = c_1c_2\dots c_n$ and candidate messages $a = a_1a_2\dots a_p$, $b = b_1b_2\dots b_q$ and outputs the number of ways those messages could be interleaved to produce the signal.

a. Identify a subproblem to solve at the ith bit of the signal c.

b. Use your subproblem to define a recurrence. Also state your base cases.

   (*hint:* try defining a two-dimensional recurrence, with each dimension corresponding to one of our candidate messages. You may not have to use the whole table...)

c. How big is your lookup table? In what order should we fill out our lookup table? How can we use the lookup table to produce a final solution?

d. Consider the signal $c = 110110$ and candidate messages $a = 101$, $b = 110$. In how many ways can these be interleaved to create c?

   Use your recurrence to fill out a lookup table.

e. Use backtracking on your lookup table to find an interleaving of messages that produces the signal c.