You will be randomly assigned groups to work on these problems in discussion section. List your group members on your worksheet and turn it in at the end of class.

**Problem 1. Texting.**

You are competing with your friends to type text messages on your smartphone as quickly as possible. Here are the rules: you use two thumbs for texting and they start out on the bottom left and bottom right keys of the keyboard. To type a character, you move either thumb from its current key to the key you need to press, and it takes time equal to the distance between the keys. You can assume the following:

- The keyboard has keys labeled \( \{1, 2, \ldots, k\} \) and there is a function \( \text{dist}(i, j) \) to calculate the distance between two keys \( i \) and \( j \). (To visualize this, you may want imagine the digits 1 through 9 arranged on a standard numeric keypad).
- Your left thumb starts on key \( a \), and your right thumb starts on key \( b \). (For example, on the 9-digit numeric keypad, \( a = 7 \) is the bottom left key, and \( b = 9 \) is the bottom right key.)
- You can press any key with either thumb
- Both thumbs can rest on the same key if necessary
- The characters to by typed are \( c_1 c_2 \cdots c_n \), where \( c_i \in \{1, 2, \ldots, k\} \) is the \( i \)th key to push

Design an algorithm that finds the fastest way to type the message. In other words, your algorithm needs to decide which thumb to use to type each character, and it should minimize the total distance moved by your two thumbs. Try to use \( O(nk^3) \) time.

**Example.** Imagine the 9-digit numeric keypad where your thumbs start at \( a = 7 \) and \( b = 9 \), with input message \( c_1 c_2 c_3 = 589 \). The solution “left, right, left” would look like this:

0 Left/right thumbs start at 7/9

1. Left thumb moves from 7 to \( c_1 = 5 \). Time = \( \text{dist}(7, 5) \). Thumbs end at 5/9.
2. Right thumb moves from 9 to \( c_2 = 8 \). Time = \( \text{dist}(9, 8) \). Thumbs end at 5/8.
3. Left thumb moves from 5 to \( c_3 = 9 \). Time = \( \text{dist}(5, 9) \). Thumbs end at 9/8.

Total time = \( \text{dist}(7, 5) + \text{dist}(9, 8) + \text{dist}(5, 9) \).
Problem 2. Longest palindromic sub-sequence.

Consider sequence of characters $a_1, a_2, \cdots a_n$ such that $a_i \in \{A, C, G, T\}$ for all $i$. A sub-sequence is any subset of these numbers taken in order, of the form $a_{i_1}, a_{i_2}, \cdots, a_{i_k}$ where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, and an palindromic sub-sequence is one which is the same whether read left to right or right to left. For instance, the sequence A, C, G, T, G, T, C, A, A, A, T, C, G has many palindromic sub-sequences, including A, C, G, C, A and A, A, A, A (on the other hand, the sub-sequence A, C, A is not palindromic). The goal is to find length of the longest palindromic sub-sequence.

(a) Define $OPT(i, j)$ to be the length of the longest palindromic sub-sequence of the sequence $a_i, a_2, \cdots a_j$. Write a recurrence and base case for $OPT(i, j)$.

(b) Write the pseudo-code for a bottom-up dynamic programming algorithm that runs in $O(n^2)$ time.