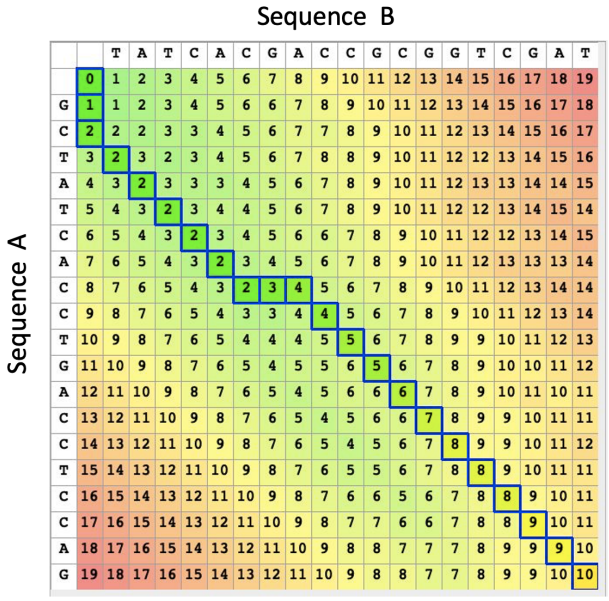
Instructions: Please note that handwritten assignments **will not be graded**. Use the provided LaTeX template to complete your homework. Please do not alter the order or spacing of questions (keep each question on its own page). When you submit to Gradescope, you must mark which page(s) correspond to each question. **You may not receive credit for unmarked questions**.
When including graphical figures, we encourage the use of tools such as graphviz or packages like tikz for simple and complex figures. However, these may be handwritten only if they are neat and legible (as defined by the grader).

List any collaborators (besides TAs or professors) here:

1. (15 points) [W10, ⋆] Edit Distance Traceback.

   Consider the following illustration of the edit distance between two gene sequences:



		T	A	T	C	A	C	G	A	C	C	G	C	G	G	T	C	G	A	T
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
G	1	1	2	3	4	5	6	6	7	8	9	10	11	12	13	14	15	16	17	18
C	2	2	2	3	3	4	5	6	7	7	8	9	10	11	12	13	14	15	16	17
T	3	2	3	2	3	4	5	6	7	8	8	9	10	11	12	12	13	14	15	16
A	4	3	2	3	3	3	4	5	6	7	8	9	10	11	12	13	13	14	14	15
T	5	4	3	2	3	4	4	5	6	7	8	9	10	11	12	12	13	14	15	14
C	6	5	4	3	2	3	4	5	6	6	7	8	9	10	11	12	12	13	14	15
A	7	6	5	4	3	2	3	4	5	6	7	8	9	10	11	12	13	13	13	14
C	8	7	6	5	4	3	2	3	4	5	6	7	8	9	10	11	12	13	14	14
C	9	8	7	6	5	4	3	3	4	4	5	6	7	8	9	10	11	12	13	14
T	10	9	8	7	6	5	4	4	4	5	5	6	7	8	9	9	10	11	12	13
G	11	10	9	8	7	6	5	4	5	5	6	5	6	7	8	9	10	10	11	12
A	12	11	10	9	8	7	6	5	4	5	6	6	6	7	8	9	10	11	10	11
C	13	12	11	10	9	8	7	6	5	4	5	6	6	7	8	9	9	10	11	11
C	14	13	12	11	10	9	8	7	6	5	4	5	6	7	8	9	9	10	11	12
T	15	14	13	12	11	10	9	8	7	6	5	5	6	7	8	8	9	10	11	11
C	16	15	14	13	12	11	10	9	8	7	6	6	5	6	7	8	8	9	10	11
C	17	16	15	14	13	12	11	10	9	8	7	7	6	6	7	8	8	9	10	11
A	18	17	16	15	14	13	12	11	10	9	8	8	7	7	7	8	9	9	9	10
G	19	18	17	16	15	14	13	12	11	10	9	8	8	7	7	8	9	9	10	10

   Answer the following questions by filling in the boxes with "insert", "delete", "match" or "substitute" as appropriate:

   (a) (2 points) If the parent pointer of a cell points to the cell *directly to the left*, then a(n) [insert] in Sequence A occurred.

   (b) (2 points) If the parent pointer of a cell points to the cell *directly above*, then a(n) [delete] in Sequence A occurred.

   (c) (2 points) If the parent pointer of a cell points to the cell *directly diagonal to the top-left*, and the values in the cells within the table are *equal* then a(n) [match] occurred.

   (d) (2 points) If the parent pointer of a cell points to the cell *directly diagonal to the top-left*, and the values in the cells within the table are *not equal* then a(n) [substitute] occurred.

(e) (7 points) Provide the edit trace for the example above in a format similar to the following:

I	A	G	O	-																
D	M	M	M	I																
-	A	G	O	G																
G	C	T	A	T	C	A	C	-	-	C	T	G	A	C	C	T	C	C	A	G
D	D	M	M	M	M	M	M	I	I	M	S	M	S	S	S	M	M	S	M	S
-	-	T	A	T	C	A	C	G	A	C	C	G	C	G	G	T	C	G	A	T

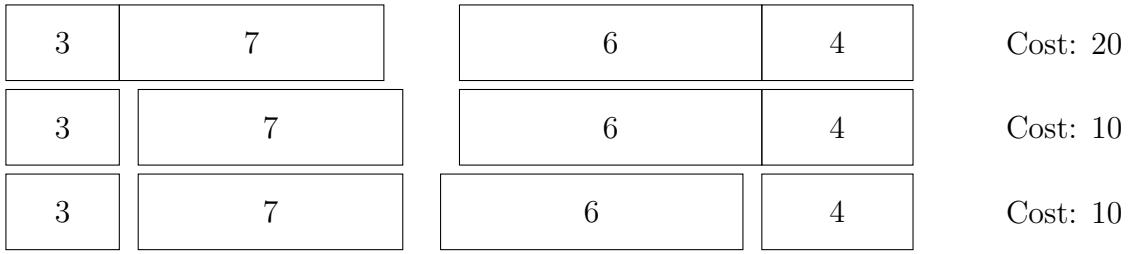
2. (50 points) [W10, ★★★★★] A certain string processing language allows the programmer to break a string, into two pieces. It costs n units of time to break a string, of n characters into two pieces, since this involves copying the old string.

   You want to break a string, A, into m pieces labeled $S_1, S_2, \ldots, S_m$ with lengths of $l_1, l_2, \ldots, l_m$, respectively. (Concatenating $S_1, S_2, \ldots, S_m$ together would form A.)

   The order in which the breaks are made can affect the total amount of time used. For example, suppose you wish to break a 20-character string after characters 3, 8, and 10 (meaning $l_1 = 3, l_2 = 5, l_3 = 2, l_4 = 10$). If the breaks are made in left-right order, then the first break costs 20 units of time, the second break costs 17 units of time, and the third break costs 12 units of time, for a total of 49 units of time. If the breaks are made in right-left order, the first break costs 20 units of time, the second break costs 10 units of time, and the third break costs 8 units of time, for a total of only 38 units of time.

   Your task is to develop a dynamic programming algorithm that takes a list of piece lengths $(l_1, \ldots, l_m)$ and determines the cheapest break cost in $\mathcal{O}(n^3)$ time.

   (a) (10 points) Provide the optimal break cost for a string of length 20 and split pieces of lengths 3, 7, 6, 4.

   Total cost: $20 + 10 + 10 = 40$

   (b) (5 points) What are the inputs to the **subproblem instance**?

   The inputs to the sub problem instance are the starting index i, the ending index j, and the length of the string n.

   (c) (5 points) Which subproblem instances are the **base cases**? What are their values?

   The base cases are when $i = j$ and the values of the base cases are 0.

   (d) (15 points) Given your answers to the above questions, develop notation and write a *formula* the **recurrence relation**.

   Recurrence relation:

$$\text{cost}(i,j) = \begin{cases} 0 & \text{if } i = j \\ \min_{i \le k < j} \left\{ \text{cost}(i,k) + \text{cost}(k+1,j) + \sum_{n=i}^{j} l_n \right\} & \text{otherwise} \end{cases}$$

   (e) (5 points) How many **dimensions** will the dynamic programming table (used in bottom-up DP) have? What variables will each dimension of your table correspond to? What is the domain of each dimension?

   There will be two dimensions in the dynamic programming table. The first dimension will correspond to the starting index i and the second dimension will correspond to the ending index j. The domain of each dimension will be $1 \le i \le n$ and $1 \le j \le n$.

(f) (10 points) What **iteration order** is required to compute the bottom-up DP table?

The first step will to set the base cases (diagonal) to 0. Then, the iteration order will be to iterate over the table in a diagonal manner. That is the first iteration will be to compute the values for $i = j$ (base case), then $i = j - 1$, $i = j - 2$, and so on until $i = 1$ and $j = n$.