Please submit one assignment per group.

Have you noticed that online searches work even if you have spelling errors in your query terms? Deciding if two strings are similar is not only important for spell-checking, but also for molecular biology. An organism's *genome* is divided up into giant linear DNA molecules know as *chromosomes*. Each chromosome can be imagined as a very long string over the alphabet $\{A, C, G, T\}$.

In the early 1970s, tow molecular biologists Needlemann and Wunsch proposed a definition of similarity which we use to this day. Consider two strings $X = x_1x_2 \dots x_m$ and $Y = y_1y_2 \dots y_n$. The sets $[m]$ and $[n]$ contain the indices of both strings. We say that (i, j) is a matching if $i \in [m]$ and $j \in [n]$ and $x_i = y_j$. We say that a matching M of these two sets is an *alignment* if there are no *crossing pairs*, meaning that if $(i, j), (i', j') \in M$, then $i < i'$ and $j < j'$. We also allow for *gaps*, represented with '-' (hyphen) which do not need to be matched.

For example, these two strings:

```
stop-
-tops
```

have the alignment $\{(2, 1), (3, 2), (4, 3)\}$.

Our definition of *similarity* will be based on finding an *optimal* alignment between X and Y, according to the following criteria. Suppose M is a given alignment between X and Y.

- First, there is a parameter $\delta > 0$ that defines a *gap penalty*. For each position of X or Y that is not matched in M, i.e., a gap, we incur a cost of δ.
- Second, for each pair of letters p, q in our alphabet, there is a *mismatch cost* of α_{pq} for lining up p with q. Thus, for each $(i, j) \in M$, we pay the appropriate mismatch cost $\alpha_{x_iy_j}$ for lining up x_i and y_j. Note that $\alpha_{pp} = 0$.
- The *cost* of M is the sum of its gaps and mismatch costs, and we seek an alignment of minimum cost.

As an example, imagine that you always forget how to spell the word "occurrence," and you spell it as "ocurrance." One possible alignment would be:

```
o-currance
occurrence
```

and another alignment would be:

```
o-curr-ance
occurre-nce
```

So the first alignment has one gap and one mismatch, while the second alignment has three gaps. Note that the first is strictly better iff $\delta + \alpha_{ae} < 3\delta$.

Design a dynamic programming solution to the minimal cost alignment. The lower the minimal cost, the more similar the two strings are.