14.2.4
Dynamic programming algorithm for edit-distance

As part of the input...

The cost of aligning a character against another character

Σ: Alphabet

We are given a **cost** function (in a table):

$$\forall \boldsymbol{b}, \boldsymbol{c} \in \Sigma \qquad \boldsymbol{COST}[\boldsymbol{b}][\boldsymbol{c}] = \text{ cost of aligning } \boldsymbol{b} \text{ with } \boldsymbol{c}.$$
$$\forall \boldsymbol{b} \in \Sigma \qquad \boldsymbol{COST}[\boldsymbol{b}][\boldsymbol{b}] = \mathbf{0}$$

$\boldsymbol{\delta}$: price of deletion of insertion of a single character

Memoizing the Recursive Algorithm (Explicit Memoization)

Input: Two strings

$A[1 \ldots m]$

$B[1 \ldots n]$

```
EditDistance(A, B)
    int   M[0..m][0..n]
    ∀i, j   M[i][j] ← ∞
    return edEMI(m, n)
```

```
edEMI( i , j )        // A[1 . . . i], B[1 . . . j]
    if M[i][j] < ∞
        return M[i][j]    // stored value

    if i = 0 or j = 0
        M[i][j] = (i + j)δ
        return M[i][j]

    m1 = δ + edEMI( i − 1 , j )
    m2 = δ + edEMI( i , j − 1)

    m3 = COST [A[i]] [B[j]]
        +   edEMI( i − 1 , j − 1 )

    M[i][j] = min(m1, m2, m3)
    return M[i][j]
```

Dynamic program for edit distance

Removing Recursion to obtain Iterative Algorithm

```
EDIST(A[1..m], B[1..n])
    int  M[0..m][0..n]
    for i = 1 to m do M[i, 0] = iδ
    for j = 1 to n do M[0, j] = jδ

    for i = 1 to m do
        for j = 1 to n do
            M[i][j] = min { COST[A[i]][B[j]] + M[i − 1][j − 1],
                          { δ + M[i − 1][j],
                          { δ + M[i][j − 1]
```

Analysis

1. Running time is $O(mn)$.

Dynamic program for edit distance

Removing Recursion to obtain Iterative Algorithm

```
EDIST(A[1..m], B[1..n])
    int   M[0..m][0..n]
    for i = 1 to m do M[i, 0] = iδ
    for j = 1 to n do M[0, j] = jδ

    for i = 1 to m do
        for j = 1 to n do
            M[i][j] = min { COST[A[i]][B[j]] + M[i − 1][j − 1],
                          { δ + M[i − 1][j],
                          { δ + M[i][j − 1]
```

Analysis

1. Running time is $O(mn)$.

Dynamic program for edit distance

Removing Recursion to obtain Iterative Algorithm

$EDIST(A[1..m], B[1..n])$
- $int \quad M[0..m][0..n]$
- for $i = 1$ to m do $M[i, 0] = i\delta$
- for $j = 1$ to n do $M[0, j] = j\delta$

- for $i = 1$ to m do
  - for $j = 1$ to n do

$$M[i][j] = \min \begin{cases} COST\left[A[i]\right]\left[B[j]\right] + M[i-1][j-1], \\ \delta + M[i-1][j], \\ \delta + M[i][j-1] \end{cases}$$

Analysis

1. Running time is $O(mn)$.
2. Space used is $O(mn)$.

THE END

...

(for now)