**Edit Distance**

The Problem: Given two sequences, how far apart are they?

**How:** What is the minimum amount of work to transform one into the other.

**Example:**

\[ x = "\text{BREAKFAST}" \]
\[ y = "\text{BLANKET}" \]

**Applications:**

- Spell checkers
- Genome sequence matching
Edit Distance

The rules:

Given: \( X[1..m] \), \( Y[1..n] \)

As each letter is processed, there options:

- Keep the letter in \( X \), it matches letter in \( Y \). Cost = 0.
  
  e.g. 'B' \( \rightarrow \) 'B'

- Replace a letter in \( X \) with a letter in \( Y \). Cost = 1.
  
  e.g. 'B' \( \rightarrow \) 'L'

- Delete a letter in \( X \). Cost = 1
  
  e.g. 'B' \( \rightarrow \) 

- Insert a letter in \( Y \). Cost = 1
  
  e.g. 'N'
Edit Distance

Example Edit

\[ \text{x: BR\_EA\_FA\_ST} \]
\[ \text{y: BL\_AN\_ET\_} \]
\[ \text{cost: 0 1 1 0 1 1 1 1 1} = 8 \]

Is this minimum?

What order is the exhaustive search to find the solution?

We can do better with dynamic programming.
What are the sub problems?
What are the edges of the DAG?
How do we process the sub problems to linearize the DAG?
How do we initialize the data?
What is the runtime?
What is the memory requirement?
Edit Distance  Sub problems

What is the best edit distance for 'B' → 'B'?

What about BR → BL?

What about BRE → BLA?

BREA → BLAN?

Is the complexity increasing?

What does BREA → BLAN depend on?

Do you see a recurrence?
Edit Distance

Full problem: $E(m,n)$ cost of full strings.

Sub problem: $E(i,j)$ cost of translating $x[i..j] \Rightarrow y[i..j]$

Recurrence:

$$E(i,j) = \min \begin{cases} 1 + E(i-1,j), & \text{Delete} \\ 1 + E(i,j-1), & \text{Insert} \\ \text{diff}(x[i..j],y[i..j]) + E(i-1, j-1), & \text{replace or keep} \end{cases}$$

$$\text{diff}(a,b) = \begin{cases} 0 & \text{if } a == b \\ 1 & \text{if } a \neq b \end{cases}$$
<table>
<thead>
<tr>
<th>Edit Distance</th>
<th>The DAG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BREAKFAST</strong></td>
<td><strong>BLANKET</strong></td>
</tr>
</tbody>
</table>

Each grid is a sub problem.
Where are the edges?
def edit_distance(X[:m], Y[:n]):
    for i = 0..m:
        E(i, 0) = i
    for j = 0..n:
        E(0, j) = j
    for i = 1..m:
        for j = 1..n:
            E(i, j) = min \{ 1 + E(i-1, j),
                           1 + E(i, j-1),
                           d(i, j) + E(i-1, j-1) \}
    return E(m, n)