Dynamic Programming Algorithms

Minimum Edit Distance

CS 336: Design and Analysis of Algorithms

How to measure distances between strings?

- **Hamming Distance:** Distance between a and b is the number of positions where a and b differ i.e.,

$$d_H(a,b) = |\{i: a_i \neq b_i\}|$$

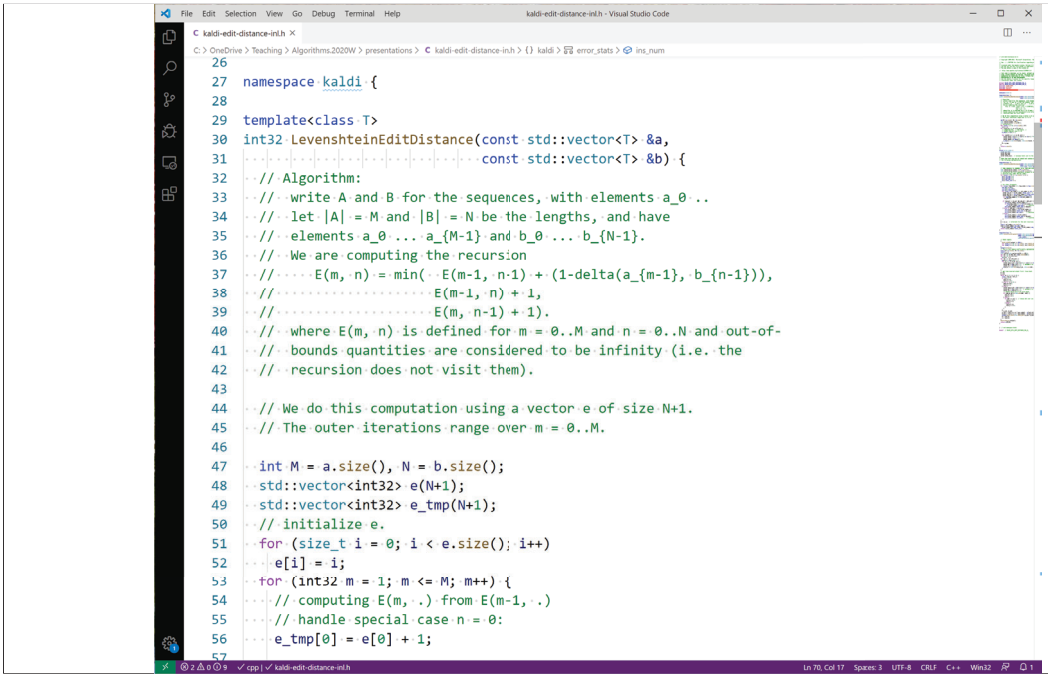
- **Pros:** Easy to compute. Works well in many applications.
- **Cons:** Doesn't work in some other applications:
  - Text editing: Typos "Hamming Distance" vs "Hmming Distance".
  - Bioinformatics: DNA mutations.

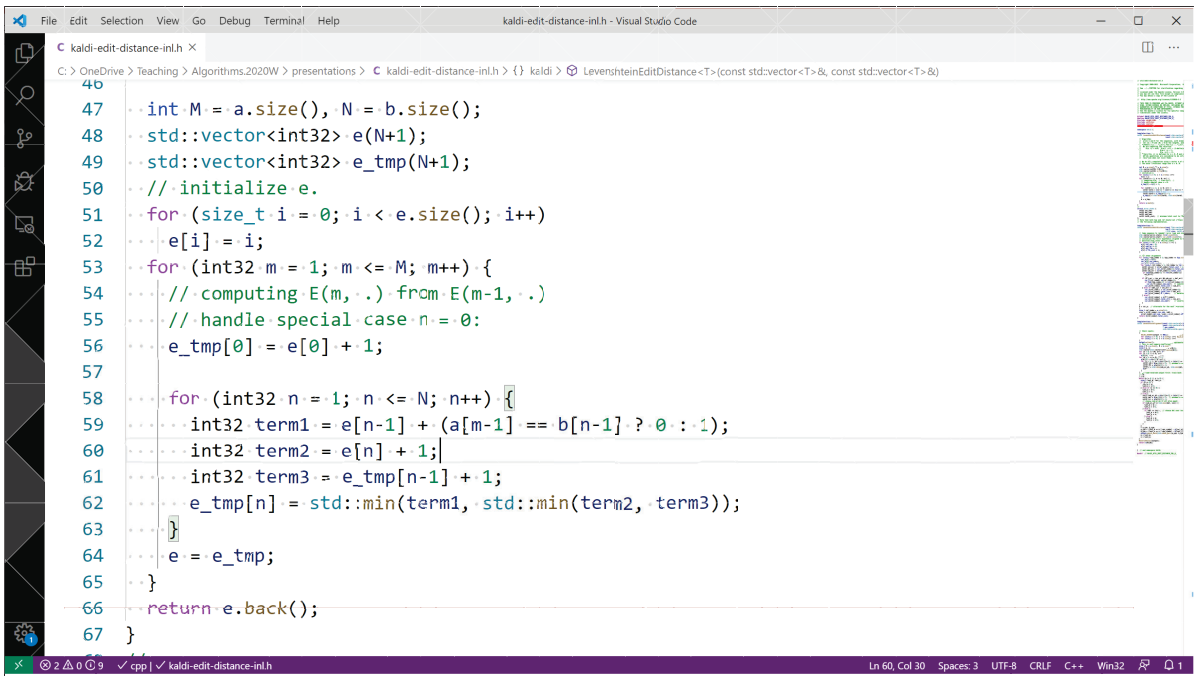
How to measure distances between strings?

- **Edit or Levenshtein distance:** The distance between $\boldsymbol{a}$ and $\boldsymbol{b}$ equals the minimum number of deletions, insertions, and substitutions required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Applications:

- Text processing: Distance from "Hamming Distance" to "Hmming Distance" equals 1.
- Bioinformatics: Edit distance is a good similarity measure for DNA strands.





Edit Distance

Given: Two strings $a_1, \dots, a_n$ and $b_1, \dots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Let $K[i,j]$ be the edit distance between prefixes $a_1, \dots a_i$ and $b_1, \dots, b_j$.

We need to find $K(n,m)$.

How to compute $K(i,j)$?

Given: Two strings $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Base case: $K[i,0] = i$ and $K[0,j] = j$.

Why?

How to compute $K(i,j)$?

Given: Two strings $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Need a recursive formula for $K[i,j]$:

Option A: $a_i = b_j$, then we don't need to change a_i.

Option B: a_i is replaced with b_j.

Option C: a_i is deleted.

Option D: b_j is inserted.

How to compute $K(i,j)$?

Given: Two strings $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Need a recursive formula for $K[i,j]$:

Option A: $a_i = b_j$, then we don't need to change a_i.

$$K[i,j] = K[i-1, j-1]$$

Option B: a_i is replaced with b_j.

Option C: a_i is deleted.

Option D: b_j is inserted.

How to compute $K(i,j)$?

Given: Two strings $a_1, \dots, a_n$ and $b_1, \dots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Need a recursive formula for $K[i,j]$:

Option A: $a_i = b_j$, then we don't need to change a_i.

Option B: a_i is replaced with b_j.

$$K[i,j] = K[i-1, j-1] + 1$$

Option C: a_i is deleted.

Option D: b_j is inserted.

How to compute $K(i,j)$?

Given: Two strings $a_1, \dots, a_n$ and $b_1, \dots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Need a recursive formula for $K[i,j]$:

Option A: $a_i = b_j$, then we don't need to change a_i.

Option B: a_i is replaced with b_j.

Option C: a_i is deleted.

$$K[i,j] = K[i-1, j] + 1$$

Option D: b_j is inserted.

How to compute $K(i,j)$?

Given: Two strings $a_1, \dots, a_n$ and $b_1, \dots, b_m$.
Goal: Find the minimum number of operations – deletions, insertions, and substitutions – required to transform $\boldsymbol{a}$ to $\boldsymbol{b}$.

Need a recursive formula for $K[i,j]$:

Option A: $a_i = b_j$, then we don't need to change a_i.

Option B: a_i is replaced with b_j.

Option C: a_i is deleted.

Option D: b_j is inserted.

$$K[i,j] = K[i, j-1] + 1$$

Algorithm for Edit Distance

for all i, j:

if $(a_i = b_j)$

a. $K[i,j] = K[i-1, j-1]$

else

$K[i,j] = \min\{$

b. $K[i-1, j-1] + 1,$

c. $K[i-1, j] + 1,$

d. $K[i, j-1] + 1\}$

Edit Distance

- The running time of the algorithm is $O(nm)$.
- The running time is almost optimal.
- In practice, people sometimes use approximate methods for computing edit distance.

Questions?