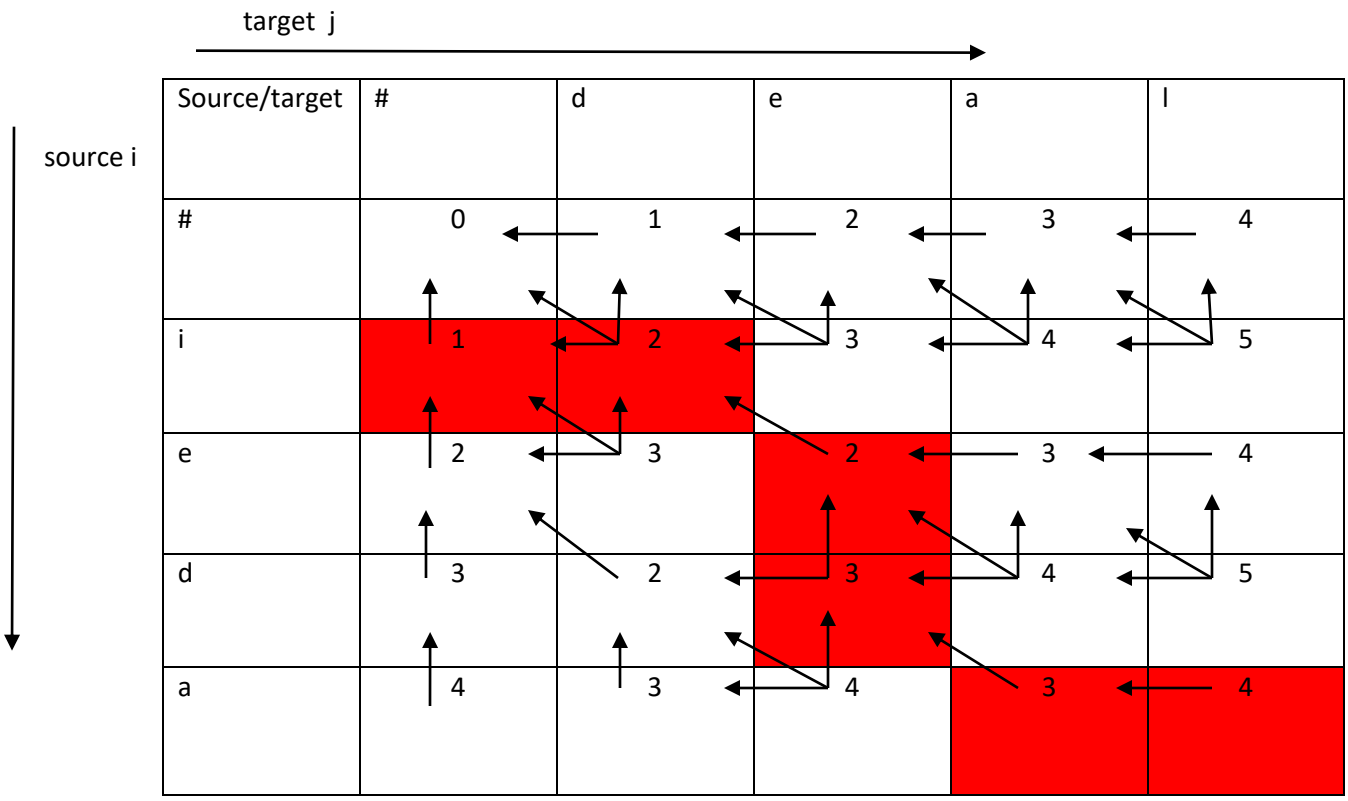
Minimum Edit Distance For two strings



Edit distance

The minimum edit distance distance between two strings

Is the minimum number of editing operations

Insertion

Deletion

Substitution

Searching for a path (sequence of edits) from the start string to the fina string:

- Initial state: the word we are transforming

- Operations: insert , delete,substitute
- Goap state: the word we are trying to get to
- Path cost: what we want to minimize : the number of edits

For two strings

- X of length m
- Y of length n

We define D(I,j)

- The edit distance between X[1 2….i] to Y[1 2…..j]
- The first i character of X and first j character of Y

Thus the edit distance between X and Y is D(n,m)

Computing Minimum Edit Distance

Dynamic Programming

- **A tabular computation of D(n,m)**
- **Solving problems by combining solutions to subproblems**
- **Bottom-up**
  - **Compute D(i,j) for small I,j**
  - **Compute larger D(I,j) based on previous computed smaller values**
  - **Compute D(i,j) for all I,j till you get to D(n,m)**

Initialization

D(i,0) =0

D(0,j)=0

Recurrence Relation:

For i = 1…m

For j=1….n

$$D(i,j)=\min \begin{cases} D(i-1,j)+1 \\ D(l,j-1)+1 \\ D(i-1,j-1)+ \begin{cases} 2; & \text{if } X(i)\neq Y(i) \\ 0; & \text{if } X(i)=Y(i) \end{cases} \end{cases}$$

Termination

D(n,m) is distance

Source→ i e d a

i e d a →delete(i)

e d a → insert(d)

d e d a →delete(d)

d e a →insert(l)

d e a l (target)

Program to implement the minimum edit distance in python.

```
n=4 m=4

target=['#',’d’,’e’,’a’,’l’]

source=['#',’i’,’d’,’e’,’a’]
a,b=5,5
dist=[[0 for i in range(a)]for j in range(b)]
for i in range(1,a):
   dist[0][i]=dist[0][i-1]+1
for j in range(1,b):
   dist[j][0]=dist[j-1][0]+1

def dellcost(s):
   return 1
def subcost(i,j):
   if(i!=j):
      return 2
   else:
      return 0
def inscost(i):
   return 1

print(dist)

for i in range(1,a):
   for j in range(1,b):
       p=i-1
       q=j-1
       dist[i][j]=min(dist[p][j] + dellcost(source[i]), dist[p][q] +
subcost(source[i],target[j]), dist[i][q] + inscost(target[j]))
print(" minimum distance matrix is :")
print(dist)
```