



//. [0-1 Knapsack Problem](https://www.geeksforgeeks.org/dynamic-programming-set-10-0-1-knapsack-problem/)

Minimum Edit Distance Algorithm

<https://www.youtube.com/watch?v=b6AGUjqIPsA>

Step1 – Array should be created with one extra length for null.

|  |  |
| --- | --- |
| ↘ Replace | ↓Remove |
| →Insert | Min (Replace, Remove, Insert) |

string str1 = "sunday", str2 = "saturday";

int[,] distanceArray = new int[m + 1, n + 1];

if (i == 0 && j == 0)

distanceArray[i, j] = 0;

else if (i== 0 && j > 0)

distanceArray[i, j] = distanceArray[i, j - 1] + 1;

else if (i > 0 && j == 0 )

distanceArray[i, j] = distanceArray[i -1, j] + 1

j

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NULL(0) | S(1) | A(2) | T(3) | U(4) | R(5) | D(6) | A(7) | Y(8) |
| NULL(0) | 0 | 1→ | 2→ | 3→ | 4→ | 5→ | 6→ | 7→ | 8→ |
| S(1) | 1↓ |  |  |  |  |  |  |  |  |
| U(2) | 2↓ |  |  |  |  |  |  |  |  |
| N(3) | 3↓ |  |  |  |  |  |  |  |  |
| D(4) | 4↓ |  |  |  |  |  |  |  |  |
| A(5) | 5↓ |  |  |  |  |  |  |  |  |
| Y(6) | 6↓ |  |  |  |  |  |  |  |  |

i

I=1 and j=0, every time when I is incremented j should be started with 0; we put j=0 logic is up.

If anyone has null values means either I or J are 0 then above condition will take care of it and it will create a above matrix.

I=1 and j=1.

Since we have extra length in loop, we compare string with 1 length -1

str1[I - 1] == str2[j - 1] {“S” == “S”} Match then took carry previous cross value.

distanceArray[I,j] = distanceArray [i-1,j-1]

Note- distanceArray[1,1] = distanceArray[0,0] which is 0, means no operation is required.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NULL(0) | S(1) | A(2) | T(3) | U(4) | R(5) | D(6) | A(7) | Y(8) |
| NULL(0) | 0 | 1→ | 2→ | 3→ | 4→ | 5→ | 6→ | 7→ | 8→ |
| S(1) | 1↓ | ↘ 0 |  |  |  |  |  |  |  |
| U(2) | 2↓ |  |  |  |  |  |  |  |  |
| N(3) | 3↓ |  |  |  |  |  |  |  |  |
| D(4) | 4↓ |  |  |  |  |  |  |  |  |
| A(5) | 5↓ |  |  |  |  |  |  |  |  |
| Y(6) | 6↓ |  |  |  |  |  |  |  |  |

Now, increment j=2, so (I = 1 and j=2)

str1[I - 1] == str2[j - 1] {“S” == “A”} not match

Took min of distanceArray (Insert, Remove and Replace) + 1;

distanceArray[I,j] = min(distanceArray[I-1,j-1]//Replace, distanceArray[I,j-1] //Insert, distanceArray[i-1,j] //Remove) + 1

distanceArray[1,2] = min(distanceArray[0,1]//Replace, distanceArray[1,1]//insert//, distanceArray[0,2]//Remove) + 1

distanceArray[1,2] = min(1,0,2) + 1; which is (0 + 1) = 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NULL(0) | S(1) | A(2) | T(3) | U(4) | R(5) | D(6) | A(7) | Y(8) |
| NULL(0) | 0 | 1→ | 2→ | 3→ | 4→ | 5→ | 6→ | 7→ | 8→ |
| S(1) | 1↓ | ↘ 0 | 1 |  |  |  |  |  |  |
| U(2) | 2↓ |  |  |  |  |  |  |  |  |
| N(3) | 3↓ |  |  |  |  |  |  |  |  |
| D(4) | 4↓ |  |  |  |  |  |  |  |  |
| A(5) | 5↓ |  |  |  |  |  |  |  |  |
| Y(6) | 6↓ |  |  |  |  |  |  |  |  |

Final array

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NULL(0) | S(1) | A(2) | T(3) | U(4) | R(5) | D(6) | A(7) | Y(8) |
| NULL(0) | 0 | 1→ | 2→ | 3→ | 4→ | 5→ | 6→ | 7→ | 8→ |
| S(1) | 1↓ | ↘ 0↖ copy from replace | 1 = 0 (min 0,1,2) +1 | 2 = 1+1 | 3=2+1 | 4 = 3+1 | 5 = 4+1 | 6 = 5+ 1 | 7 = 6+1 |
| U(2) | 2↓ | 1 | 1← | 2← | 2↖ | 3 | 4 | 5 | 6 |
| N(3) | 3↓ | 2 | 2 | 2 | 3 | 3↖ | 4 | 5 | 6 |
| D(4) | 4↓ | 3 | 3 | 3 | 3 | 4 | 3↖ | 4 | 5 |
| A(5) | 5↓ | 4 | 3 | 4 | 4 | 4 | 4 | 3↖ | 4 |
| Y(6) | 6↓ | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 3↖ |

Total operation is required Sunday to Saturday is = 3 last value of array distanceArray[6,8].

Travel bottom to up on diagnosis. Here start from bottom to read array and get operation value.

If match (no changes are required)

If distanceArray[m,n] == distanceArray[m-1,n-1] match character.

Here it keep matching until distanceArray[3,5] and distanceArray[2,4]

If not match distanceArray[3,5] != distanceArray[2,4] { 3 != 2)

Then take a look it matches with Replace or Insert. If both match then given a priority to replace.

Here replaces character str1[m-1] to str2[n-1], str1[2] = “n” replaces to str2[4] = “r”.