Tutorial 1

Anshul Choudhary(17CS10005)

17-07-2019

1 Problem Statement

$A[1..m]$ and $B[1..n]$ are two 1D arrays containing m and n integers respectively, where $m \leq n$. We need to construct a sub-array $C[1..m]$ of B such that $\sum_{i=1}^{m} \left|A[i] - C[i]\right|$ is minimized.

2 Recurrences

Lets denote M[i][j] as the minimum value of $\sum_{j=1}^{i} \left|A[j] - C[j]\right|$ when array $A[1..i]$ and $B[1..j]$ are considered where $i >= j$, $i <= m$ and $j <= n$.

$$M[i][j] = \begin{cases} |A[i] - B[j]| \quad if\ i = 1\ and\ j = 1 \\ \\ min\{|A[i] - B[j]|, M[i][j-1]\}; \quad if\ i = 1\ and\ j! = 1 \\ \\ INT\_MAX; \quad if\ i > j \\ \\ min\{|A[i] - B[j]| + M[i-1][j-1], M[i][j-1]\} \quad otherwise \end{cases}$$

Here M[n][m] is the final ans.

3 Algorithm

```
int M[n][m]
for i = 1 to n do
    for j = 1 to m do
        if i == 1 then
            if j==1 then
                M[i][j] = |A[i] − B[j] ;
            end
            else
                M[i][j] = min{|A[i] − B[j], M[i][j − 1]} ;
            end
        end
        else if i > j then
            M[i][j] = INT_MAX ;
        end
        else
            M[i][j] = min{|A[i] − B[j]| + M[i − 1][j − 1], M[i][j − 1]} ;
        end
    end
end
char B[m]
i=m
j=n
while(j!=0)
{
  while(i > 0ansM[j][i] == M[j][i − 1])
  {
   i–
  }
  C[m − i] = B[j][i]
  j–
}
```

4 Demonstration

Lets take an example ,

$$A = [4\ 5\ 8\ 6\ 7]$$

$$B = [2\ 4\ 3\ 1]$$

So the matrix M created is,

$$M = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ X & 3 & 3 & 3 & 3 \\ X & X & 8 & 6 & 6 \\ X & X & X & 13 & 12 \end{bmatrix}$$

By calculating C[] from above matrix, we get,

$$C = [4\ 5\ 6\ 7]$$

5 Time and space complexities

From the above pseudo code we can calculate Time Complexity as,
(i) outer for loop will run n times —-> O(n)
(ii) inner for loop will run m times —-> O(m)
So total time will be,

$$T = O(n) * O(m)$$

$$TimeComplexity = O(n * m)$$

Since we are using a 2D matrix of size m x n, the Space Complexity will be,

$$SpaceComplexity = O(n * m)$$