Tutorial - 1

Kousshik Raj (17CS30022)

23 - 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 $\sum_{i=1}^{m} \big|A[i] - C[i]\big|$ is minimized.

2 Recurrences

Let $A[1..i]$ and $B[1..j]$ be a portion of the two given 1D arrays containing i and j integers respectively, where $i \leq j$. We have to create a sub-array of B, $C[1..i]$, so as to minimize the value of $dp(i,j) = \sum_{x=1}^{i} \big|A[x] - C[x]\big|$.

$$dp(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ \infty & \text{if } i < j \\ \min\left(|A[i] - B[j]| + c(i-1, j-1), c(i, j-1)\right) & \text{else} \end{cases}$$

The first case corresponds to the array A being empty, for which the sum of the absolute differences boil down to 0.

The case $j < i$ corresponds to the case where the array B is smaller than the array A, which is not within the range of constraints of the problem and hence an infinite cost has been assigned to it.

The final case has been explained in the next section.

The minimum value of the sum of absolute differences obtained by choosing the optimal sub-array of B, C, to our original problem statement will be given by $dp(m,n)$.

3 Algorithm

For this recurrence we are following a bottom up approach, i.e, we start with $i = 0$ & $j = 0$. Let us assume we have the optimal sub-array $C[1..i]$ of $B[1..j]$

that produces the smallest value of $dp(i, j)$ (the sum of absolute differences), for some i & j.

We can encounter only two possible scenarios, either we choose the last element of the array B as an element of the optimal sub-array C, or not.

If the last element of the array B, $B[j]$, is chosen, then it should be the last element of the array C as it is an sub-array, hence, $C[i] = B[j]$. The above situation boils down to the problem for the arrays $A[1..i-1]$, $B[1..j-1]$. Thus, for this case, $dp(i, j) = |A[i] - B[j]| + dp(i-1, j-1)$

If $B[j]$ is not present in the array C, then the last element of B is not part of the optimal sub-array C and can be safely ignored from consideration and hence the problem then repeats for the arrays $A[1..i]$, $B[1..j-1]$ and $C[1..i]$. Thus, for this case, $dp(i, j) = dp(i, j-1)$

The minimum sum of absolute differences can only arise from either of these two cases, and hence the minimum of the two cases is chosen. The optimal sub-array C can be obtained from the array dp.

4 Demonstration

Let $A = \boxed{2}\boxed{7}\boxed{2}$, $B = \boxed{5}\boxed{3}\boxed{6}\boxed{8}$

For our algorithm,
$A = \boxed{2}\boxed{7}\boxed{2}$, $B = \boxed{5}\boxed{3}\boxed{6}\boxed{8}$

After initialization,

Algorithm 1 BestSubArray

```
Input
A[1..m] of length m
B[1..n] of length n

Output
C[1..m] of length m

procedure BestSubArray(A[1..m], B[1..n])
    Create empty array C[1..m], dp[0..m][0..n]

    for i ← 0 to m do                ▷ Initializing the dp array with base cases
        for j ← 0 to n do
            if i = 0 then
                dp[i][j] ← 0
            else if j < i then
                dp[i][j] ← ∞

    for i ← 1 to m do                ▷ Calculating the dp array using the recurrence
        for j ← i to n do
            dp[i][j] ← min(|A[i] − B[j]| + dp[i − 1, j − 1], dp[i, j − 1])

    row ← m
    column ← n
    while row ≠ 0 do
        a = |A[row] − B[column]| + dp[row − 1][column − 1]
        b = dp[row][column − 1]
        if a ≤ b then
            C[row] ← B[column]
            row ← row − 1
        column ← column − 1
    return C
```

$$dp = \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline \infty & - & - & - & - \\ \hline \infty & \infty & - & - & - \\ \hline \infty & \infty & \infty & - & - \\ \hline \end{array}$$

$$\begin{aligned} dp[1][1] &= \min\left(|A[1] - B[1]| + dp[0][0], dp[1][0]\right) \\ &= \min\left(|-3| + 0, \infty\right) \\ &= 3 \\ dp[1][2] &= \min\left(|A[1] - B[2]| + dp[0][1], dp[1][1]\right) \\ &= \min\left(|-1| + 0, 3\right) \\ &= 1 \\ &\vdots \\ dp[3][4] &= \min\left(|A[3] - B[4]| + dp[2][3], dp[3][3]\right) \\ &= \min\left(|-6| + 2, 11\right) \\ &= 8 \end{aligned}$$

After completion,

$$dp = \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline \infty & 3 & 1 & 1 & 1 \\ \hline \infty & \infty & 7 & 2 & 2 \\ \hline \infty & \infty & \infty & 11 & 8 \\ \hline \end{array}$$

Final sub-array retrieved from dp

$C = \begin{array}{|c|c|c|}\hline 3 & 6 & 8 \\ \hline \end{array}$

5 Time and space complexities

Lines	Time Complexity
7	$O(1)$
8..13	$O(m \cdot n)$
14..16	$O(m \cdot n)$
17..25	$O(n)$
Total	$O(m \cdot n)$

Lines	Space Complexity
7	$O(m) + O(m \cdot n)$
8..13	$O(1)$
14..16	$O(1)$
17..25	$O(1)$
Total	$O(m \cdot n)$