Tutorial 1

Akash Tiwari (17CS10003)

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 Assumptions/Intro

Let the **expression to be minimized** be denoted as **f**.Let us consider a matrix T of dimensions $n * m$.
We will only use a part of the matrix such that $T[i][j]$ is usable if and only if $i \geq j$.
$T[i][j]$ represents the minimum value of f if only first i entries of $B[1..n]$ and only first j entries of $A[1..m]$ are taken.

3 Recurrences

The value of $T[i][j]$ can be **founded by recurrence** by the expression:
$T[i][j] = min[\mid B[i] - A[j] \mid + T[i-1][j-1], T[i-1][j]]$
Note/Observations:

- min takes the value $\mid B[i] - A[j] \mid + T[i-1][j-1]$ when we are including the element $B[i]$ in C.
- min takes the value $T[i-1][j]$ when we are not including the element $B[i]$ in C.
- Here , $i \geq 1$, $j \geq 1$, $i > j$. Base cases will be handled separately.

4 Base Cases

- $T[0][0] = \mid A[0] - B[0] \mid$
- For all i where $i \geq 1$ and $i \leq n$, $T[i][0] = min(\mid B[i] - A[0] \mid, T[i-1][0])$

- For all i where $i \geq 1$ and $i \leq m$, $T[i][i] = T[i-1][i-1] + \mid B[i] - A[i] \mid$

5 Algorithm

Explanation:

- First we find the base cases of the matrix using rules defined in base cases.
- Then using the recurrence relation we fill the viable entries (according to rule in assumptions).
- The T[n-1][m-1]th element will be the minimum sum that can be obtained.
- we start from the T[n-1][m-1]th element and backtrack, in the process we determine whether to select B[i] or not using the observations in Recurrences.
- We do edge case handling and the final result is obtained.

Pseudocode

```
// assuming existence of mod function which returns absolute value
// assuming existence of a min function which returns min of 2 values

//defining matrix and output answer
int T[n][m], C[m]
// assuming input arrays are inputted
int A[m], B[n]
//base case no 1.
T[0][0]= mod(A[0] - B[0])
// loop for assigning base case no 2 and 3
for (i=1;i<n;i++)
    //base case no 2
    T[i][0] = min(mod(B[i]-A[0]), T[i-1][0])
    // base case no 3
    T[i][i] = T[i-1][i-1] + mod(A[i] -B[i])
//Filling the matrix according to recurrence
for(i=2;i<=n;i++)
    for(j=1;j<i && j<m;j++)
        //recurrence relation
        T[i][j] =min(mod(B[i]-A[j])+T[i-1][j-1], T[i-1][j])
//initialising i and j for next iteration
i=n-1,j=m-1
// backtracking to find array element
while(j>0)
    if T[i][j]==mod(B[i]-A[j])+T[i-1][j-1]
    //this being true means, B[i] will be included in the sequence
```

```
        C[j]=B[i]
        i--
        j--
    else
    //B[i] is not included in the sequence
        i--
//finding first element of c
while(i>=0)
    if(T[i][0]==mod(B[i]-A[0]))
        //if found, then break
        C[0] = B[i]
        break
    else
        // else move back up
        i--
    // code complete
```

6 Demonstration

Test case 1: let $A = 2, 7, 2$ and B be $B = 5, 3, 6, 8$.
Then possible cases of B are:

- $C = 5, 3, 6$ then the value of f is $\mid 2 - 5 \mid + \mid 7 - 3 \mid + \mid 2 - 6 \mid = 11$
- $C = 5, 3, 8$ then the value of f is $\mid 2 - 5 \mid + \mid 7 - 3 \mid + \mid 2 - 8 \mid = 13$
- $C = 5, 6, 8$ then the value of f is $\mid 2 - 5 \mid + \mid 7 - 6 \mid + \mid 2 - 8 \mid = 10$
- $C = 3, 6, 8$ then the value of f is $\mid 2 - 3 \mid + \mid 7 - 6 \mid + \mid 2 - 8 \mid = 8$

clearly the **expected answer** is : $C = 3, 6, 8$
Now following the algorithm:
T matrix initially(x **indicates not assigned**)

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & x & x & x \\ 1 & x & x & x \\ 2 & x & x & x \\ 3 & x & x & x \end{bmatrix}$$

T matrix after the assigning base cases

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & 3 & x & x \\ 1 & 1 & 7 & x \\ 2 & 1 & x & 10 \\ 3 & 1 & x & x \end{bmatrix}$$

T matrix after going through recurrence loop

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & 3 & x & x \\ 1 & 1 & 7 & x \\ 2 & 1 & 2 & 10 \\ 3 & 1 & 2 & 8 \end{bmatrix}$$

We find the min value of f at $T[4][2](8) = 8$, which is **consistent with our expected value 8**.
Now **we backtrack** from $T[3][2]$ to find the C:
1st iteration of backtracking:
Since $T[3][2] = T[2][1]+ \mid B[3] - A[2] \mid$ holds true $C[2] = B[3] = 8$
2nd iteration of backtracking:
Since $T[2][1] = T[1][0]+ \mid B[2] - A[1] \mid$ holds true $C[1] = B[2] = 6$
1st iteration on reaching j=0, for finding C[0]:
Since $T[1][0] =\mid B[1] - A[0] \mid$ holds true $C[0] = B[1] = 3$
So finally, $C = 3, 6, 8$ which is consistent with our expected answer.

Test case 2: let $A = 1, 8, 4$ and B be $B = 7, 5, 9, 3, 2$.
Then possible cases of B are:

- $C = 7, 5, 9$ then the value of f is $\mid 7 - 1 \mid + \mid 5 - 8 \mid + \mid 9 - 4 \mid = 14$
- $C = 7, 5, 3$ then the value of f is $\mid 7 - 1 \mid + \mid 5 - 8 \mid + \mid 3 - 4 \mid = 10$
- $C = 7, 5, 2$ then the value of f is $\mid 7 - 1 \mid + \mid 5 - 8 \mid + \mid 2 - 4 \mid = 11$
- $C = 7, 9, 3$ then the value of f is $\mid 7 - 1 \mid + \mid 9 - 8 \mid + \mid 3 - 4 \mid = 8$
- $C = 7, 9, 2$ then the value of f is $\mid 7 - 1 \mid + \mid 9 - 8 \mid + \mid 2 - 4 \mid = 9$
- $C = 7, 3, 2$ then the value of f is $\mid 7 - 1 \mid + \mid 3 - 8 \mid + \mid 2 - 4 \mid = 13$
- $C = 5, 9, 3$ then the value of f is $\mid 5 - 1 \mid + \mid 9 - 8 \mid + \mid 3 - 4 \mid = 6$
- $C = 5, 9, 2$ then the value of f is $\mid 5 - 1 \mid + \mid 9 - 8 \mid + \mid 2 - 4 \mid = 7$
- $C = 5, 3, 2$ then the value of f is $\mid 5 - 1 \mid + \mid 3 - 8 \mid + \mid 2 - 4 \mid = 11$
- $C = 9, 3, 2$ then the value of f is $\mid 9 - 1 \mid + \mid 3 - 8 \mid + \mid 2 - 4 \mid = 15$

clearly the **expected answer** is : $C = 5, 9, 3$
Now following the algorithm:
T matrix initially(x indicates not assigned)

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & x & x & x \\ 1 & x & x & x \\ 2 & x & x & x \\ 3 & x & x & x \\ 4 & x & x & x \end{bmatrix}$$

T matrix after the assigning base cases

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & 6 & x & x \\ 1 & 4 & 9 & x \\ 2 & 4 & x & 14 \\ 3 & 2 & x & x \\ 4 & 1 & x & x \end{bmatrix}$$

T matrix after going through recurrence loop

$$\begin{bmatrix} indexa/indexb & 0 & 1 & 2 \\ 0 & 6 & x & x \\ 1 & 4 & 9 & x \\ 2 & 4 & 5 & 14 \\ 3 & 2 & 5 & 6 \\ 4 & 1 & 5 & 6 \end{bmatrix}$$

We find the min value of f at $T[4][2] = 6$, which is **consistent with our expected value** 6.
Now **we backtrack** from $T[4][2]$ to find the C:
1st iteration of backtracking:
Since $T[4][2] \neq T[3][1]+ \mid B[3] - A[2] \mid$, as they are not equal, we move 1 cell above without assigning anything to $C[2]$
2nd iteration of backtracking:
Since $T[3][2] = T[2][1]+ \mid B[3] - A[2] \mid$ holds true $C[2] = B[3] = 3$
3rd iteration of backtracking:
Since $T[2][1] = T[1][0]+ \mid B[2] - A[1] \mid$ holds true $C[1] = B[2] = 9$
1st iteration on reaching j=0, for finding C[0]:
Since $T[1][0] =\mid B[1] - A[0] \mid$ holds true $C[0] = B[1] = 5$
So finally, $C = 5, 9, 3$ which is **consistent with our expected answer**.

7 Time and space complexities

Time complexity (capital M,N used for better readability)
Time complexity for assigning base cases=O(N)+O(M)
Time complexity for filling the matrix = O(MN)(Since each element is evaluated in O(1) time)
Time complexity for backtracking = O(N)
Time complexity for evaluating C[0] after backtracking = O(N)
Therefore **overall Time complexity**(adding since loops are independently running) = **O(MN)**
Space complexity(capital M,N used for better readability)
Space complexity for matrix= O(MN)
Space complexity for output array= O(M)
Overall space complexity = O(MN)