Indian Institute of Technology, Kharagpur
Computer Science and Engineering Department

Autumn 2019-2020
Algorithms II (CS31005)

Note:
You are advised to use LaTeX for document preparation.
wiki link for LaTeX: `https://en.wikibooks.org/wiki/LaTeX`
You can also see `https://www.latex-tutorial.com/tutorials/`
See **LaTeX in Ubuntu** in the next page.

Tutorial Problem T1 [17-07-2019—23-07-2019]

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

1. Develop the recurrences needed for DP, with clear arguments. (50 marks)
2. Design the algorithm and write the pseudo-code. (20 marks)
3. Demonstrate your algorithm on a few input instances. (20 marks)
4. Derive the time and space complexities of your algorithm. (10 marks)

Example

Let $A = \boxed{2 \mid 7 \mid 2}$ and $B = \boxed{5 \mid 3 \mid 6 \mid 8}$.

Possible cases:

1. $C = \boxed{3 \mid 6 \mid 8}$
   $A = \boxed{2 \mid 7 \mid 2}$
   sum $= 1 + 1 + 6 = 8$.
2. $C = \boxed{5 \mid 6 \mid 8}$
   $A = \boxed{2 \mid 7 \mid 2}$
   sum $= 3 + 1 + 6 = 10$.
3. $C = \boxed{5 \mid 3 \mid 8}$
   $A = \boxed{2 \mid 7 \mid 2}$
   sum $= 3 + 4 + 6 = 13$.
4. $C = \boxed{5 \mid 3 \mid 6}$
   $A = \boxed{2 \mid 7 \mid 2}$
   sum $= 3 + 4 + 4 = 11$.

So, here the solution is $C = \boxed{3 \mid 6 \mid 8}$.

Solution of T1

Optimal substructure Define $f(m,n)$ as the required minimum over $A[1..m]$ and $B[1..n]$. Let $C[m] = B[k]$. As C has m elements taken from n elements of B, we have $m \le k \le n$. Hence, we get

$$f(m,n) = \min_{m \le k \le n} \Big\{ f(m-1, k-1) + \big|A[m] - B[k]\big| \Big\}. \tag{1}$$

Overlapping subproblems Define $f(i,j)$ as the required minimum over $A[1..i]$ and $B[1..j]$ for $1 \le i \le j$. Generalizing Eq. 1 with inclusion of base cases, we get

$$f(i,j) = \begin{cases} \text{undefined} & \text{if } i > j \\ \min\limits_{1 \le k \le j} \Big\{ \big|A[1] - B[k]\big| \Big\} = \min \Big\{ f(1, j-1), \big|A[1] - B[j]\big| \Big\} & \text{if } i = 1 \\ \min\limits_{i \le k \le j} \Big\{ f(i-1, k-1) + \big|A[i] - B[k]\big| \Big\} & \text{otherwise.} \end{cases} \tag{2}$$

Time and space complexities To compute $f[i][j]$ we need $O(j-i+1)$ time due to the innermost **for** loop with k as the loop variable. Considering the outermost and its next **for** loops, the total runtime is $T(m,n) = O\Big(\sum\limits_{i=1}^{m} \sum\limits_{j=i}^{n} j - i + 1\Big) = O\Big(\sum\limits_{i=1}^{m} \frac{1}{2}(n-i)(n-i+1)\Big) = O\Big(\sum\limits_{i=1}^{m} n^2\Big) = O(mn^2)$. The array needs $O(mn)$ space, and that's the overall space complexity of the algorithm.

An Improved Solution of T1

Optimal substructure Define $f(m,n)$ as the required minimum over $A[1..m]$ and $B[1..n]$. The element $B[n]$ is either not in C or the last element of C. Hence, we get

$$f(m,n) = \min \Big\{ f(m, n-1), f(m-1, n-1) + \big|A[m] - B[n]\big| \Big\}. \tag{3}$$

Overlapping subproblems Define $f(i,j)$ as the required minimum over $A[1..i]$ and $B[1..j]$ for $1 \le i \le j$. Generalizing Eq. 3 with inclusion of base cases, we get

$$f(i,j) = \begin{cases} \text{undefined} & \text{if } i > j \\ \big|A[1] - B[1]\big| & \text{if } i = j = 1 \\ \min \Big\{ f(1, j-1), \big|A[1] - B[j]\big| \Big\} & \text{if } i = 1 \text{ and } j > 1 \\ \min \Big\{ f(i, j-1), f(i-1, j-1) + \big|A[i] - B[j]\big| \Big\} & \text{otherwise.} \end{cases} \tag{4}$$

Examples

Input: $A = 2, 4, 7$ and $B = 3, 2, 6, 9$.
Output: $f[3][4] = 4$.

	3	2	6	9
2	1	0	0	0
4	–	3	2	2
7	–	–	4	4

Input: $A = 2, 7, 3, 5$ and $B = 7, 5, 7, 2, 4, 5, 2$.
Output: $f[4][7] = 4$.

	7	5	7	2	4	5	2
2	5	3	3	0	0	0	0
7	–	7	3	3	3	2	2
3	–	–	11	4	4	4	3
5	–	–	–	14	5	4	4

Algorithm 1: Closest-sub-sequence

```
for(i=0; i<m; i++) // initialization
  for(j=0; j<n; j++)
    f[i][j] = INT_MAX; // in <limits.h>

f[0][0] = abs(a[0]-b[0]);

for(j=1; j<n; j++) // 1st row
  f[0][j] = min((f[0][j-1]), (abs(a[0]-b[j])));

for(i=1; i<m; i++){ // other rows
  for(j=i; j<n; j++){
    f[i][j] = min((f[i][j-1]), (f[i-1][j-1]+abs(a[i]-b[j])));}}

printf("\nAns.= %d\n", f[m-1][n-1]);
```

Time and space complexities To compute $f[i][j]$ we need $O(1)$ time, and hence the total time complexity is $O(mn)$. The space complexity is $O(mn)$ needed for the 2D array.

LaTeX in Ubuntu

1. To install LaTeX in Ubuntu, use the following command:
   `sudo apt-get install texlive-full`

2. Open an editor like `gedit` or `kile`.

3. Create a file using that editor, say with the name `a.tex`, with the following content.

```
\documentclass{article}
\title{Tutorial 1}
\date{17-07-2019}
\author{Your name (and roll number)}

\begin{document}
  \maketitle

  \section{Problem Statement}
    $A[1..m]$ and $B[1..n]$ are two 1D arrays containing $m$ and $n$ integers
    respectively, where $m\le n$.
    We need to construct a sub-array $C[1..m]$ of $B$ such that
    $\sum\limits_{i=1}^{m} \big|A[i]-C[i]\big|$ is minimized.

  \section{Recurrences}

    Text ..........

  \section{Algorithm}

    Text ..........

  \section{Demonstration}

    Text ..........

  \section{Time and space complexities}

    Text ..........

\end{document}
```

4. Compilation command: `pdflatex a.tex`
   It will create the output file named `a.pdf`.

5. Open `a.pdf` in some pdf viewer.
   It will look as shown in the next page!

a.pdf

Tutorial 1

Your name (and roll number)

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

Text ....

3 Algorithm

Text ....

4 Demonstration

Text ....

5 Time and space complexities

Text ....