

ASSIGNMENT 3

DFS LAB

Deadline: October 10, 2017

Each question carries 20 marks.

Q1. **Matrix Sort.** You are given an $n \times m$ matrix A . The entries of A are integers in $[0, 9]$ (i.e., single digit positive numbers). Assume that the rows and columns of the matrix are indexed starting from 0. Write a program to transform A to another matrix B , also of size $n \times m$, such that the following conditions hold.

- The numbers in each row of B appear in non-decreasing order from left to right.
- For any two rows i and j of B with $i < j$,

$$\sum_{k=0}^{m-1} 10^k A[i][m-k-1] \leq \sum_{k=0}^{m-1} 10^k A[j][m-k-1]. \quad (1)$$

For example, if

$$A = \begin{bmatrix} 5 & 6 & 9 & 4 \\ 3 & 1 & 3 & 2 \\ 1 & 5 & 3 & 8 \\ 2 & 5 & 4 & 7 \\ 3 & 5 & 1 & 1 \\ 4 & 6 & 5 & 7 \end{bmatrix}, \text{ then } B = \begin{bmatrix} 1 & 1 & 3 & 5 \\ 1 & 2 & 3 & 3 \\ 1 & 3 & 5 & 8 \\ 2 & 4 & 5 & 7 \\ 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 9 \end{bmatrix}.$$

Input file format:

Number of rows, n

Number of columns, m

Elements of the matrix A ($n \times m$ integers in a single line, separated by spaces)

Output format:

Elements of the matrix B in the same format as above ($n \times m$ integers in a single line, separated by spaces)

NOTE: The matrix entries may be too large for the summation in Equation (1) to be computed.

Q2. **Pattern Occurrence.** Consider a string S (sequence of characters) of length n . S consists only of lowercase English letters. Write a program to find the number of distinct 4-tuples $\langle i, j, k, l \rangle$ (not necessarily disjoint) such that $1 \leq i < j < k < l \leq n$ and $S[i] == S[k]$, as well as $S[j] == S[l]$.

Examples:

Input: $S = \text{abab}$

Output: 1

Explanation: Only tuple is $\langle i = 1, j = 2, k = 3, l = 4 \rangle$.

Input: $S = \text{ababa}$

Output: 2

Explanation: Tuples are $\langle i = 1, j = 2, k = 3, l = 4 \rangle$ and $\langle i = 2, j = 3, k = 4, l = 5 \rangle$.

Input Format:

S provided as a command-line argument

- Q3. You are given n integers i_1, i_2, \dots, i_n . The numbers need to be added. At a time, you can only add two numbers. Suppose that the cost of adding two numbers is equal to the sum of the numbers. Write a program that takes i_1, i_2, \dots, i_n as command line arguments, and prints the minimum total cost of finding $\sum_{j=1}^n i_j$ by adding two numbers at a time.

Example:

Input: 4 3 2 6

Output: 29

Explanation:

Method 1 for finding the sum of the above 4 numbers:

Step 1. Add 4 and 3 (cost = 7); available numbers are 7, 2, 6.

Step 2. Add 7 and 2 (cost = 9); available numbers are 9, 6.

Step 3. Add 9 and 6 (cost = 15).

Total cost = $7+9+15= 31$.

Method 2 for finding the sum of the above 4 numbers:

Step 1. Add 2 and 3 (cost = 5); available numbers are 4, 5, 6.

Step 2. Add 4 and 5 (cost = 9); available numbers are 9, 6.

Step 3. Add 9 and 6 (cost = 15).

Total cost = $5+9+15= 29$.

After trying all possible combinations, it turns out that the minimum over all combinations is 29 (Method 2 above).