Assignment 8

Design & Analysis of Algorithms Lab

March 15, 2022

Edit Distance Problem Find the total minimum cost of edits (Insert, Delete, and Replace operations) required to convert str1 into str2. Insert, Delete, and Replace operations have cost 1 unit, 1 unit, and 2 units respectively.

- cost[i][j] denotes the cost for converting s[1..i] to t[1..j]
- cost[i][j-1] + 1 denotes the insertion of the j-th character of t to t[1..j-1] that we already have got from s[1..i]
- cost[i-1][j] + 1 denotes the deletion of the i-th character of s since we already have formed t[1..j] from s[1..i-1]

  In addition, you need to print out the sequence of insert, delete, and replace operations on str1 that converts it into str2.

  Example: str1 = "tea" str2 = "eat"

  Minimum Edit Distance for converting tea to eat is 2

  Deletion of t at 0

  Insertion of t at 2

Matrix Chain Multiplication Problem Given a sequence of n matrices $\{M_0, M_1, M_2, \ldots, M_{n-1}\}$ with r_i and c_i are the number of rows and columns respectively of the matrix M_i, find the most efficient way to multiply these matrices together. Assume that the input is given correctly, i.e., $M_i.c = M_{i+1}.r$ for all $0 \leq i \leq n-2$. The problem is not actually to perform the multiplications, but merely to decide in which order to perform the multiplications.

We have many options to multiply a chain of matrices because matrix multiplication is associative. In other words, no matter how we parenthesize the product, the result will be the same. For example, if we had four matrices M_0, M_1, M_2, and M_3, we would have 5 ways:

$((M_0M_1)M_2)M_3 = (M_0(M_1M_2))M_3 = (M_0M_1)(M_2M_3) = M_0((M_1M_2)M_3) = M_0(M_1(M_2M_3))$

However, the order in which we parenthesize the product affects the number of basic multiplications needed to compute the product. For example, suppose M_0 is a 10 × 30 matrix, M_1 is a 30 × 5 matrix, and M_2 is a 5 × 60 matrix. Then,

$(M_0M_1)M_2$ = (10×30×5) + (10×5×60) = 1500 + 3000 = 4500 operations.

$M_0(M_1M_2)$ = (30×5×60) + (10×30×60) = 9000 + 18000 = 27000 operations.

Write a C/C++ program that returns the minimum number of multiplications needed to multiply the chain. Also print the order in which the matrices should be multiplied.

Egg Dropping Puzzle A *critical floor* is defined as the highest floor from which an egg can be dropped without breaking. Given a K-storied building ($K \leq 100$), and N eggs ($N \leq 6$), write a C/C++ program that finds out the critical floor with minimum number of egg dropping attempts. Note that if an egg is not broken for i-th floor, then it won't be broken while being dropped from 1 to (i-1)-th floor. Also, if an egg is broken for i-th floor, then it will be broken while being dropped from (i+1)-th to K-th floor.

Example: N = 2 K = 6 Attempts = 3

Maximum Three Sum Subsequence Given an array A[] of n positive integers, write a C/C++ program to find the maximum sum of the subsequence that can be formed by picking up the elements in the array such that the subsequence contains exactly one element from every subarray of three consecutive elements.

Example 1:

n = 6 A[] = {1, 2, 3, 4, 5, 6}

{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, 6} are possible subarrays of length 3

Output: 9 {3 + 6}

Example 2:

n = 6 A[] = {1, 2, 3, 6, 5, 4}

{1, 2, 3}, {2, 3, 6}, {3, 6, 5}, {6, 5, 4} are possible subarrays of length 3

Output: 9 {3 + 6}

Example 3:

n = 6 A[] = {3, 2, 1, 4, 5, 5}

{3, 2, 1}, {2, 1, 4}, {1, 4, 5}, {4, 5, 5} are possible subarrays of length 3

Output: 7 {2 + 5} [First 5 has been picked]

Submission Instruction:

File Name: A8_RollNo.c/cpp
Email to: *pds2016autumn@gmail.com* with **subject line**: A8_RollNo