Bharatiya Vidya Bhavan's

SARDAR PATEL INSTITUTE OF TECHNOLOGY

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri (W), Mumbai – 400 058.

Experiment No. 4

Aim – To implement dynamic algorithms

Details – Dynamic Programming is a technique in computer programming that helps to efficiently solve a class of problems that have overlapping sub-problems and optimal substructure property. If any problem can be divided into sub-problems, which in turn are divided into smaller sub-problems, and if there are overlapping among these sub-problems, then the solutions to these sub-problems can be saved for future reference. The approach of solving problems using dynamic programming algorithm has following steps:

1. Characterize the structure of an optimal solution.
2. Recursively define the value of an optimal solution.
3. Compute the value of an optimal solution, typically in a bottom-up fashion.
4. Construct an optimal solution from computed information.

Problem Definition & Assumptions – The aim of this experiment is two-fold. First, it finds the efficient way of multiplying a sequence of k matrices (called ***Matrix Chain Multiplication***) using Dynamic Programming. The chain of multiplication M_1 x M_2 x M_3 x M_4 x...x M_k may be computed in $(2N)!/((N+1)!\,N!) = \binom{2N}{N}/(N+1)$ ways due to associative property where $N = k - 1$ of matrix multiplication. Second, it compares regular matrix multiplication which has complexity of $O(n^3)$ and Strassen's Matrix Multiplication which has complexity of $O(n^{2.81})$ using Divide and Conquer.

Consider the optimization problem of efficiently multiplying a randomly generated sequence of 10 matrices (M_1, M_2, M_3, M_4 ,..., M_{10}) using Dynamic programming approach. The dimension of these matrices are stored in an array p[i] for i = 0 to 9, where the dimension of the matrix M_i is (p[i-1] x p[i]). All p[i] are randomly generated and they are between 15 and 46. For example, $p[0..10] = (23, 20, 25, 45, 30, 35, 40, 22, 15, 29, 21)$. All ten matrices are generated randomly and each matrix value can be between 0 and 1. Determine following values of ***Matrix Chain Multiplication (MCM)*** using Dynamic Programming:

1) m[1..10][1..10] = Two dimension matrix of optimal solutions (No. of multiplications) of all possible matrices M_1... M_{10}
2) c[1..9][2..10] = Two dimension matrix of optimal solutions (parenthesizations) of all combinations of matrices M_1...M_{10}
3) the optimal solution (i.e.parenthesization) for the multiplication of all ten matrices M_1x M_2x M_3xM_4 x...x M_{10}

Find the running time of 10 matrices using regular matrix multiplication and Strassen's Matrix Multiplication as a trivial sequence i.e. ((((((((M_1x M_2)x M_3) xM_4 x...x M_{10}) and the sequence of matrix multiplication suggested by ***Matrix Chain Multiplication*** in Step No. 3

Links for Understanding basic concepts of MCM:

1. Youtube MCM video link - https://www.youtube.com/watch?v=prx1psByp7U
2. Reading resource link - https://www.javatpoint.com/matrix-chain-multiplication-example

---

Input –

1) All p[i] for i=0 to 9 are randomly generated and they are between 15 and 46. 2) All ten matrices are generated randomly and each matrix value can be between 0 and 1.

Output –

1) m[1..10][1..10] = Two dimension matrix of optimal solutions (No. of multiplications) of all possible matrices M_1... M_{10}
2) c[1..9][2..10] = Two dimension matrix of optimal solutions (parenthesizations) of all combinations of matrices M_1...M_{10}
3) The optimal solution (i.e.parenthesization) for the multiplication of all ten matrices M_1x M_2x M_3xM_4 x...x M_{10}
4) Print the time required to multiply ten matrices using four combinations as discussed above.

Submission –

1) C/C++ source file of implementation
2) Verified output for the written source code with multiple inputs (four combinations)
3) One page report of Exp. 4 (PDF file)