CALIFORNIA STATE UNIVERSITY, FRESNO
DEPARTMENT OF COMPUTER SCIENCE

Class:	**Algorithms & Data Structures**	Semester:	**Fall 2023**

Points		Document author:	**Pratham Aggarwal**
		Author's email:	**Pratham_aggarwal@mail.fresnostate.edu**
		Laboratory number:	**Lab 9**

1. Statement of Objectives

The objective of this experiment is to use dynamic programming to implement and examine the time complexity of solving matrix chain multiplication. Understanding the challenge of parenthesizing a chain of matrices to reduce the number of scalar multiplications and creating a dynamic programming solution are part of the scope. The importance is in the optimization of matrix chain multiplication, a common operation in computer science and numerical computation. Among the lab's major accomplishments is the writing of program that determines the optimal parenthesizing and least cost for a given chain of matrices.

2. Experimental Procedure

The experimental procedure includes implementing the dynamic programming solution for matrix chain multiplication. Initializing matrices to record minimum multiplication costs and parenthesizing indices, iterating through various parenthesizations, and computing the optimal solution are all part of the approach. A recursive function (printParenthesis) is used to print the optimal parenthesizing, and a main function (matrixChainOrder) is used to find the minimum cost. To show the program's functioning, sample inputs are provided.

3. Analysis

Output:

Sample Input 1: [10,20,30,40,30]
Optimal Parenthesization is : (((AB)C)D)
Optimal Cost is : 30000

Sample Input 2: [4,10,3,12,20,7]
Optimal Parenthesization is : ((AB)((CD)E))
Optimal Cost is : 1344

Sample Input 3: [10,20,80,2,16,13]
Optimal Parenthesization is : ((A(BC))(DE))
Optimal Cost is : 4276

The dynamic programming solution for matrix chain multiplication finds the best parenthesization while minimizing the number of scalar multiplications. This is accomplished by the algorithm considering all alternative parenthesizations and selecting the one with the lowest cost. The solution has a time complexity of O(n^3), where 'n' is the number of matrices in the chain.

The provided outputs demonstrate the algorithm's efficiency in optimizing matrix multiplication for various input conditions. The optimal parenthesizations provide a strategic sequence for multiplying matrices, which results in the lowest total cost of scalar multiplications. When dealing with a large number of matrices, this dynamic programming approach is very useful because it greatly reduces computational cost when compared to simple multiplication methods.

Screenshot at the end of the report.

4. Encountered Problems

While writing a program, I experienced difficulties in understanding the recursive structure of the optimal parenthesization function (printParenthesis). Also, I was having issues with some part of code, I couldn't figure out how to write that certain part. There were also starting problems with array indices and logic, but I solved them with the help of online tools and tutoring.

5. Conclusions

In conclusion, the lab provided useful insights into optimizing matrix chain multiplication using dynamic programming. The implemented program successfully finds the best parenthesization and lowest cost for a given chain of matrices. The most important takeaways are a better understanding of dynamic programming concepts, solving problems skills, and the practical application of algorithmic optimization techniques.

6. References

Class notes