
Homework 2

Patrick Indri

May 5, 2019

CHAIN MATRIX MULTIPLICATION

The `code/` folder contains the implementation of the Chain Matrix Multiplication algorithm. `chain.c` implements a bottom up approach to determine the best parenthesization for the chain multiplication of a series of n matrices. Denoting such chain multiplication as $A_{1\dots n}$ the matrix **M** is defined as matrix where $M[i][j]$ is the minimum number of scalar products needed to compute $A_{i\dots j}$. **S** denotes the matrix whose entries $S[i][j]$ record the value for the best split of $A_{i\dots j}$. **MatrixChain** computes both matrices, printing and returning **S**. **PrintParens** exploits the info in **S** to print the optimal solution.

Listing 1: Chain Multiplication Computation

```
1 Mult(i, j) {  
2   if (i == j) return A[i];  
3   else {  
4       k = S[i,j];           // Identifies best split.  
5       X = Mult(i,k);        // X = A[i]...A[k].  
6       Y = Mult(k+1, j);     // Y = A[k+1]..A[j].  
7       return XY;           // Multiply matrices X and Y.  
8   }  
9 }  
10 }
```

A general approach to actually compute the optimal solution has been implemented: the structure of such a general approach [1] is presented in Lst. 1.

The implementation has been tested on an instance of 6 matrices: the chain matrix algorithm has been compared with a naive, sequential multiplication.

Listing 2: Output of the code for the test matrix chain

```

1  1      1      3      3      3
2  0      2      3      3      3
3  0      0      3      3      3
4  0      0      0      4      5
5  0      0      0      0      5
6
7 Printing optimal solution:      ((A1(A2A3))((A4A5)A6))
8
9 NAIVE SOLUTION
10 Execution time: 0.000746s
11 Number of operations: 40500
12
13 CHAIN SOLUTION
14 Execution time: 0.000324s
15 Number of operations: 15125
16
17 Operations opt/naive: 0.373457
18
19 Is the result the same? 1

```

The code outputs the matrix S and the optimal parenthesization. On this test instance, the chain matrix algorithm shows a clear improvement in execution time and number of operation. Comparing the two results element by element, the correctness of the computation has been verified as well.

(maybe) TODO: perform a benchmark with more instances.

REFERENCES

- [1] R. B. Muhammad. *Matrix Chain Multiplication Problem*. 18.03.2010. URL: <http://personal.kent.edu/%7Ermuhamma/Algorithms/MyAlgorithms/Dynamic/chainMatrixMult.htm>.