Algorithm Theory - Assignment 4

Téo Bouvard

March 11, 2020

Problem 1

a) We compute the m-table and the s-table according to the MATRIX-CHAIN-ORDER procedure, with a slight modification : rather than replacing a cell value if the computational cost is lower, we replace it when the cost is higher. By doing this, we find the product order which maximizes the number of scalar multiplications. Note that all indices have been fixed to start at 0 for consistency reasons. This means that matrices are A_0 to A_4.

Table 1: m-table

0	15750	18000	21000	43875
	0	2625	6000	17625
		0	750	4500
			0	1250
				0

Table 2: s-table

	0	1	1	0
		1	1	1
			2	3
				3

By reconstructing the optimal solution, we get the optimal parenthesization is $(A_0(A_1((A_2A_3)A_4)))$ for a total cost of 43875 multiplications.

b) The result have been checked with the code given in the python file attached. The only modifications (apart from the zero-index fixes) are :

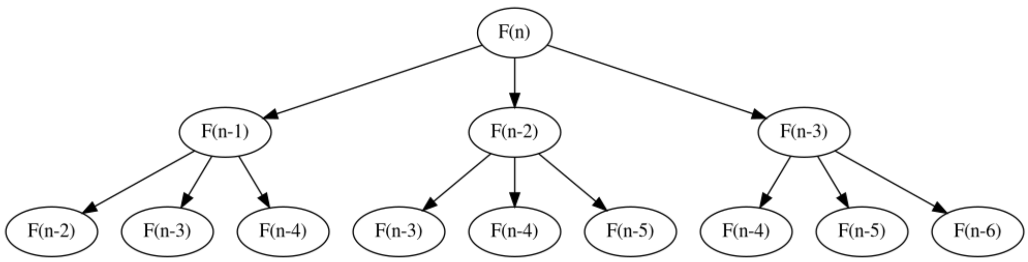
- Replacing the initialization of a cell value from infinity to zero. $m[i][j] = \infty \rightarrow m[i][j] = 0$.
- Replacing the operator in the update condition. $q < m[i][j] \rightarrow q > m[i][j]$.

Problem 2

a) A simple recursive algorithm could be used to compute the values of the series. A possible implementation is given in the function F-DAQ below.

F-DAQ(n)
  if $n < 3$
    return n
  return F-DAQ$(n-1) *$ F-DAQ$(n-2) + (n-3) *$ F-DAQ$(n-3)$

b) The subproblem graph for this algorithm is given below.



We can see that this approach is highly inefficent as subproblems are solved multiple times in the recursive tree. The time complexity can be defined recursively as T(n) = T(n-1) + T(n-2) + T(n-3), which leads to a complexity in the order of $\mathcal{O}(3^n)$.