ADw6&7

Name : **Peter Garamvoelgyi** Student ID: **2018280070**

**CLRS 15.2-1. Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is. (pg. 378)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | **0** | |  |  |  |  |  |
| **2** | - | | **0** |  |  |  |  |
| **3** | - | | - | **0** |  |  |  |
| **4** | - | | - | - | **0** |  |  |
| **5** | - | | - | - | - | **0** |  |
| **6** | - | | - | - | - | - | **0** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | 0 | | **150 k=1** |  |  |  |  |
| **2** | - | | 0 | **360 k=2** |  |  |  |
| **3** | - | | - | 0 | **180 k=3** |  |  |
| **4** | - | | - | - | 0 | **3000 k=4** |  |
| **5** | - | | - | - | - | 0 | **1500 k=5** |
| **6** | - | | - | - | - | - | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | 0 | | 150 k=1 | **330 k=2** |  |  |  |
| **2** | - | | 0 | 360 k=2 | **330 k=2** |  |  |
| **3** | - | | - | 0 | 180 k=3 | **930 k=4** |  |
| **4** | - | | - | - | 0 | 3000 k=4 | **1860 k=4** |
| **5** | - | | - | - | - | 0 | 1500 k=5 |
| **6** | - | | - | - | - | - | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | 0 | | 150 k=1 | 330 k=2 | **405 k=2** |  |  |
| **2** | - | | 0 | 360 k=2 | 330 k=2 | **2430 k=2** |  |
| **3** | - | | - | 0 | 180 k=3 | 930 k=4 | **1770 k=4** |
| **4** | - | | - | - | 0 | 3000 k=4 | 1860 k=4 |
| **5** | - | | - | - | - | 0 | 1500 k=5 |
| **6** | - | | - | - | - | - | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | 0 | | 150 k=1 | 330 k=2 | 405 k=2 | **1655 k=4** |  |
| **2** | - | | 0 | 360 k=2 | 330 k=2 | 2430 k=2 | **1950 k=2** |
| **3** | - | | - | 0 | 180 k=3 | 930 k=4 | 1770 k=4 |
| **4** | - | | - | - | 0 | 3000 k=4 | 1860 k=4 |
| **5** | - | | - | - | - | 0 | 1500 k=5 |
| **6** | - | | - | - | - | - | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **j** | | | | | | |
|  |  | **1** | | **2** | **3** | **4** | **5** | **6** |
| **i** | **1** | 0 | | 150 k=1 | 330 k=2 | 405 k=2 | 1655 k=4 | **2010 k=2** |
| **2** | - | | 0 | 360 k=2 | 330 k=2 | 2430 k=2 | 1950 k=2 |
| **3** | - | | - | 0 | 180 k=3 | 930 k=4 | 1770 k=4 |
| **4** | - | | - | - | 0 | 3000 k=4 | 1860 k=4 |
| **5** | - | | - | - | - | 0 | 1500 k=5 |
| **6** | - | | - | - | - | - | 0 |

**The best parenthesization is**

**… with cost 2010.**

**CLRS 15.3-3. Consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize, the number of scalar multiplications. Does this problem exhibit optimal substructure? (pg. 389)**

“*A problem exhibits optimal substructure if an optimal solution to the problem contains within it optimal solutions to subproblems.*” (CLRS pg. 379)

Let us prove that this problem has the optimal substructure property by using the “cut-and-paste” technique.

Let us suppose that, for a given input, the algorithm gives us the optimal solution

… consisting of the two sub-solutions

The cost (that is to be maximized in this case) is given by the formula

Let us assume the sub-solutions are not optimal, i.e.

By replacingwith, the cost of the final solutionbecomes

As, we can see that

… which is a contradiction aswas said to be an optimal solution. Thus, the subproblems must be optimal.

**CLRS 25.2-4. As it appears above, the Floyd-Warshall algorithm requiresspace, since we computefor. Show that the following procedure, which simply drops all the superscripts, is correct, and thus onlyspace is required. (pg. 699).**

Thevalues depend on

… i.e. only values from the previous () matrix. Thus, it is trivial that keeping only two matrices (last and current) at any time is sufficient. The space requirement is.

The fact that having one matrix is sufficient is less trivial, as the values

… might change before our update of , making it incorrect.

However, let us notice that values in row and columnin fact do not change:

This means that in every iteration of, recent updates ofwill not affect subsequent updates ().

Thus, keeping only a single matrix, the algorithm remains correct. This results in a non-asymptotic reduction in space requirements. The asymptotic bound remains.