Text

Description automatically generated

Name : Amudheswaran Sankaranarayanan

UBID: 50454389

UBName: amudhesw

Box and whisker chart

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *j ->* **opt[I,j]**  *i ↓* | 1` | 2 | 3 | 4 | 5 |
| 1 | **0** | **120** | **300** | **444** | **612** |
| 2 |  | **0** | **240** | **432** | **656** |
| 3 |  |  | **0** | **480** | **756** |
| 4 |  |  |  | **0** | **336** |
| 5 |  |  |  |  | **0** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *j ->* **π[I,j]**  *i ↓* | 1` | 2 | 3 | 4 | 5 |
| 1 |  | **1** | **2** | **3** | **4** |
| 2 |  |  | **2** | **3** | **4** |
| 3 |  |  |  | **3** | **3** |
| 4 |  |  |  |  | **4** |

MINIMUM COST IS **612**

Multiplication order is **((((A1 A2) A3) A4 )A5)**

Diagram

Description automatically generated

Soln

* Choose the current node, add its weight, and find the max weighted cell of its grandchildren
* Skip the current node’s weight from being added, explore that node’s immediate children, and find their max weighted cell.
* The nodes in the tree are calculated once, hence its runtime is confined to O(n)

Text

Description automatically generated

Text, letter

Description automatically generated

Soln

The dp table **memo** is initialized to 1 as that would be the smallest subsequence to ever exist, and for every element that is lesser than the element at **i** the subsequence length increasesby 1 to incorporate that element.

Then the dp table **memo** is again updated to see if the addition of the element or the pre-existing subsequence is longer up till that index position **i.**

This algo runs for **O(n^2)** as the loops run for n (input size) times and for each iteration, it re-checks the maximal length of the subsequence. But as it is memorized the values are not recalculated from scratch.

Text

Description automatically generated