Assignment 2

**Dynamic Programming Approach Explanation**:

* We use two matrices **m** and **s** to store the minimum number of scalar multiplications (**m)** and the split points (**s**) for the optimal parenthesization.
* **m[i][j]** represents the minimum scalar multiplications required to multiply matrices from **i** to **j**.
* **S[i][j]** stores the index **k** where the split should occur to minimize the scalar multiplications.

**Initialization**:

* We start by initializing **m[i][j]** for all **i** as a single matrix requires no multiplication.

**Recurrence Relation**:

* We iterate through different chain lengths and subproblems, filling in the **m** matrix.
* For each chain length and subproblem, we compute the cost of parenthesizing at various split points **k** and update **m[i][j]** if a better split is found.

**Dynamic Programming Approach Explanation**:

* We use two matrices **m** and **s** to store the minimum number of scalar multiplications (**m)** and the split points (**s**) for the optimal parenthesization.
* **m[i][j]** represents the minimum scalar multiplications required to multiply matrices from **i** to **j**.
* **S[i][j]** stores the index **k** where the split should occur to minimize the scalar multiplications.

**Initialization**:

* We start by initializing **m[i][j]** for all **i** as a single matrix requires no multiplication.

**Recurrence Relation**:

* We iterate through different chain lengths and subproblems, filling in the **m** matrix.
* For each chain length and subproblem, we compute the cost of parenthesizing at various split points **k** and update **m[i][j]** if a better split is found.