| | |
|---|---|
| **NAME:** | **Sanskar Kamble** |
| **UID:** | **2021300054** |
| **SUBJECT** | **Design and Analysis of Algorithm** |
| **EXPERIMENT NO :** | **04** |
| **DATE OF PERFORMANCE** | **06/03/2023** |
| **DATE OF SUBMISSION** | **12/03/2023** |
| **AIM:** | **To implement matrix chain multiplication and also to compute its time complexity.** |
| **THEORY:** | **Matrix chain multiplication** (or the **matrix chain ordering problem**) is an optimization problem concerning the most efficient way to multiply a given sequence of matrices. The problem is not actually to *perform* the multiplications, but merely to decide the sequence of the matrix multiplications involved. The problem may be solved using dynamic programming.<br><br>There are many options because matrix multiplication is associative. In other words, no matter how the product is parenthesized, the result obtained will remain the same.<br><br>It follows the following algorithm:<br><br>• Take the sequence of matrices and separate it into two subsequences.<br>• Find the minimum cost of multiplying out each subsequence.<br>• Add these costs together, and add in the cost of multiplying the two result matrices.<br>• Do this for each possible position at which the sequence of matrices can be split, and take the minimum over all of them. |

**Code:**

```
#include <stdio.h>
#include <limits.h>
// Matrix Ai has dimension p[i-1] x p[i]
// for i = 1 . . . n
int MatrixChainOrder(int p[], int i, int j)
{
   if (i==j)
      return 0;
   int k;
   int min = INT_MAX;
   int count;

   for (k = i; k < j; k++)
   {
      count = MatrixChainOrder(p, i, k)+
MatrixChainOrder(p, k + 1, j)+ p[i - 1] * p[k] * p[j];

      if (count < min)
         min = count;
   }

   // Return minimum count
   return min;
}
int main()
{
   int n;
   printf("Enter the number of elements:\n");
   scanf("%d",&n);
   int arr[n];
   printf("Enter the elements:\n");
  for(int i=0;i<n;i++)
  scanf("%d",&arr[i]);

   printf("Minimum number of multiplications is %d ",
        MatrixChainOrder(arr, 1, n-1));
   getchar();
   return 0;
}
```

**Output:**

```
Enter the number of elements:
5
Enter the elements:
3
4
5
2
6
Minimum number of multiplications is 100

...Program finished with exit code 0
Press ENTER to exit console.
```

**Conclusion:** By performing the above experiment I have successfully understood about the concept of matrix chain multiplication as well as its implementation.