Solution for Question (1)

Given, Dimensions of matrices:

XXXX

Let, optimal multiplications to multiply matricesthrough

So, to find fewest number of multiplications for provided matrices in question we need to find .

and, dimension of

So, ………….. eq (1)

Now,

, , ,

Also,

To calculate value of in eq(1) we also need to find and.

……………..eq(2)

Substituting values we calculated above, we get:

Similarly, calculating :

…………….eq(3)

Substituting these values we calculated in eq(1), we get:

Therefore, *fewest number of multiplications needed to find the resulting matrix* is **2856**.

Looking at which min values in eq (1), (2) and (3) gave us fewest number of multiplications:

For was fromand

And min value in was from and

Therefore, *the optimal order of multiplying the four matrices is*

Solution for Question (2)

The algorithm given in the question **will not always minimize** the number of multiplications necessary to multiply the matrix chain .

To prove let’s take three matrices: and with dimensions XXX respectively. Let’s also assume that .

So, according to the algorithm, the order of matrix multiplication which will require fewest number of multiplications would be =

Multiplications required in the above order of matrix multiplications is =

If we would have done matrix multiplications in order , then multiplications required will be =

If the proposed algorithm always work, then:

Rearranging terms in above equation, we get:

……….eq(i)

Now, , because we assumed that . But need not to be true always as there is no constraint applied on relationship of and

*Therefore eq(i) does not hold true*.

Example: let . This holds our assumption of .

Following the proposed algorithm order of multiplication will be =

Number of multiplications in above order is = ( …..eq(ii)

If we would have done multiplication in order:

Number of multiplications is = (, which is less than number of multiplications in eq(ii).

*This shows that proposed algorithm does not work.* ***Hence proved****.*

Solution for Question (3)

So,

Calculate using set of one node:

Calculate using set of two nodes:

………….eq(i)

………….eq(ii)

………….eq(iii)

Finally, to calculate length of optimal tour:

……eq(iv)

Substituting values we calculated above, we get:

Therefore, length of the optimal tour for the Traveling Salesman problem is =

Now, to determine the node ordering in the optimal tour we need to look at which were the min value in eq(i), (ii), (iii) and (iv). This gives us:

, and

Therefore, node ordering in the optimal tour is:

Solution for Question (4)

“Special Node Set (SNS)” of a graph contains nodes that do not share an edge between them. SNS with max cardinality is “Largest Special Node Set (LSNS)”.

Following is a **high-level description of algorithm to find LSNS** of a tree:

1) In a tree, at a node , let:

largest independent node set of tree rooted at .

2) Suppose we know the largest independent set of all subtrees below node . Then if we want to calculate there are two possibilities:

(i) is not in :

then is union of the where is children of .

(ii) is in :

then is union of and , where is grandchildren of .

**Recurrence relation** for this algorithm can be written as:

Time complexity of above recurrence relation can be analyzed by the number of comparisons each node is involved in. Now for each node , above algorithm processes node only three times:

(i) When processing node .

(ii) When processing node ’s parent.

(iii) When processing node ’s grandparent.

Since each node is processed constant number of times, **time complexity of the algorithm to find LSNS is** , where is number of nodes in the tree.

However, **this** **algorithm cannot be used to find LSNS of a graph which is not a tree** because this algorithm depends on processing children and grandchildren of a node, but in a cyclic graph this would cause the algorithm to be caught in an infinite loop. As base condition for this recurrence relation is for a node to be ‘NULL’ or a ‘Leaf’, in a cyclic graph these base conditions would not be satisfied.