Dynamic Programming

Matrix Chain Multiplacation

Given a list of dimensions of compatible matrices, calculate the most efficient ordering of multiplying them.

To do this we will create a $n - 1 * n - 1$ matrix, C, where n is the number of dimensions given to us. Each cell of matrix C denoted by C[i, j] describes the most efficient way to multiply matrices i through j. The goal is to calculate the cell [1, n-1] as this will hold the smallest number of multiplications needed to multiply all the compatible matrices. Note, n is the number of dimensions, so $n - 1$ is the number of matrices being multiplied. We will also store the matrices that were multiplied behind the scenes so we can retrace them for the final solution.

The following is the algorithm to do so:

1. Create a $n - 1 * n - 1$ matrix where n in the number of dimensions given to you.
   -- Note: we only need half of the matrix that we are creating to calculate the solution.
2. Initialize the diagonal values to 0
3. Calculate all adjacent matrices being multiplied starting with [1,2] then [2,3], etc
4. Proceed with matrices 2 apart starting with [1,3] then [2,4], etc
5. Continue with matrices 3 apart etc until all matrices being multipled together.
6. To calculate the cell C[i,j] when 3 matrices or more are being multiplied together. We must consider the min value of the choices from k = i...j-1
   We evaluate C[i,k] + C[k+1,j] + p(i-1) *p(k)* p(j) where p(x) is the xth element in the list of dimensions.
7. We will also store the choosen min ordering for the cell C[i,j] so we can retrace the solution later.
8. The final solution will be at [1,n-1]

The tables below demonstrate the progress of the algorithm.

Out[ ]:

	1	2	3	4	5
1	0	0	0	0	0
2	-	0	0	0	0
3	-	-	0	0	0
4	-	-	-	0	0
5	-	-	-	-	0

```
Value for cell 1, 2
c[1,2] = p0*p1*p2 = 30*60*170 = 306000

Value for cell 2, 3
c[2,3] = p1*p2*p3 = 60*170*45 = 459000

Value for cell 3, 4
c[3,4] = p2*p3*p4 = 170*45*160 = 1224000

Value for cell 4, 5
c[4,5] = p3*p4*p5 = 45*160*30 = 216000

Values considered for cell 1,3
c[1,1] + c[2,3] +  p0*p1*p3 = 0 + 459000 + 30*60*45 = 540000
c[1,2] + c[3,3] +  p0*p2*p3 = 306000 + 0 + 30*170*45 = 535500

Values considered for cell 2,4
c[2,2] + c[3,4] +  p1*p2*p4 = 0 + 1224000 + 60*170*160 = 2856000
c[2,3] + c[4,4] +  p1*p3*p4 = 459000 + 0 + 60*45*160 = 891000

Values considered for cell 3,5
c[3,3] + c[4,5] +  p2*p3*p5 = 0 + 216000 + 170*45*30 = 445500
c[3,4] + c[5,5] +  p2*p4*p5 = 1224000 + 0 + 170*160*30 = 2040000

Values considered for cell 1,4
c[1,1] + c[2,4] +  p0*p1*p4 = 0 + 891000 + 30*60*160 = 1179000
c[1,2] + c[3,4] +  p0*p2*p4 = 306000 + 1224000 + 30*170*160 = 2346000
c[1,3] + c[4,4] +  p0*p3*p4 = 535500 + 0 + 30*45*160 = 751500

Values considered for cell 2,5
c[2,2] + c[3,5] +  p1*p2*p5 = 0 + 445500 + 60*170*30 = 751500
c[2,3] + c[4,5] +  p1*p3*p5 = 459000 + 216000 + 60*45*30 = 756000
c[2,4] + c[5,5] +  p1*p4*p5 = 891000 + 0 + 60*160*30 = 1179000

Values considered for cell 1,5
c[1,1] + c[2,5] +  p0*p1*p5 = 0 + 751500 + 30*60*30 = 805500
c[1,2] + c[3,5] +  p0*p2*p5 = 306000 + 445500 + 30*170*30 = 904500
c[1,3] + c[4,5] +  p0*p3*p5 = 535500 + 216000 + 30*45*30 = 792000
c[1,4] + c[5,5] +  p0*p4*p5 = 751500 + 0 + 30*160*30 = 895500
```

	1	2	3	4	5
1	0	306000	535500	751500	792000
2	-	0	459000	891000	751500
3	-	-	0	1224000	445500
4	-	-	-	0	216000
5	-	-	-	-	0

```
Final solution:
((A_1 * A_2)(A_3))(A_4 * A_5)
```

The final table above shows the least number of multiplications needed in the top right cell:

The solution is 792,000 multiplications which corresponds to matrix order $((A_1 * A_2)(A_3))(A_4 * A_5)$

Complexity

Time complexity: $O(n^3)$ where n is the number of dimensions

- You must fill $\frac{n^2}{2}$ cells which can take $O(n)$ work per cell

Space complexity: $O(n^2)$