CS29003 Algorithms Laboratory
Assignment 3: Complexity of Algorithms

General instruction to be followed strictly

1. Do not use any global variable unless you are explicitly instructed so.
2. Do not use Standard Template Library (STL) of C++.
3. Use proper indentation in your code and comment.
4. **Name your file as `<roll_no>_<assignment_no>`. For example, if your roll number is 14CS10001 and you are submitting assignment 3, then name your file as `14CS10001_3.c` or `14CS10001_3.cpp` as applicable.**
5. **Write your name, roll number, and assignment number at the beginning of your program.**
6. Make your program as efficient as possible. Follow best practices of programming.
7. Submit your program on Moodle before deadline. Submissions by email or any other means will NOT be considered for evaluation.

Suppose the cost of multiplying two matrices $A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times p}$ on a particular computer be mr^2p. Suppose we need to compute the product of three matrices $A_1 \in \mathbb{R}^{5 \times 7}$, $A_2 \in \mathbb{R}^{7 \times 3}$, and $A_3 \in \mathbb{R}^{3 \times 9}$. There are two ways to compute $A_1 \times A_2 \times A_3$: $(A_1 \times A_2) \times A_3$ and $A_1 \times (A_2 \times A_3)$. The first way incurs a cost of 1140 where as the second way incurs a cost of 2772. Hence, the first way incurs minimum cost for this example. Can you think of how many ways are there to compute product of n matrices?

In this assignment, you will compute the minimum-cost way to compute product of n matrices $A_1, \ldots, A_n$. The input is $n+1$ positive integers, say $i_1, \ldots, i_{n+1}$, where the dimension of A_j is $i_j \times i_{j+1}$ for every $j \in \{1, \ldots, n\}$.

Here is a recursive approach to the problem. For a given input and r and s with $1 \leqslant r \leqslant s \leqslant n$, let us define $D[r, s]$ to be the minimum cost to compute the product of $A_r, \ldots, A_s$. The following formulas are easy to verify.

$$D[r, r] = 0, \forall r \in \{1, \ldots, n\}$$

$$D[r, s] = \min \left\{ D[r, t] + D[t + 1, s] + i_r i_{t+1}^2 i_{s+1} : r \leqslant t < s \right\}$$

The minimum cost to compute the product of $A_1, \ldots, A_n$ is $D[1, n]$.

Part I

Write a recursive function implementing the above algorithm. What is the running time of the above algorithm?

Part II

Design a polynomial-time algorithm for this problem. You are allowed to use $\mathcal{O}(n^2)$ space. Hint: why does the recursive algorithm in part I take long time? Can you make the algorithm in part I iterative? For obtaining full marks, your algorithm should run in $\mathcal{O}(n^2)$ time and $\mathcal{O}(n^2)$ space.

Implement the two methods in parts I and II as two separate functions. Write their time complexity in comments preceding the function definitions.

Finally, output the time spent by each method to compute the minimum cost solution. You may use the following code snippet for that.

```
#include <time.h>
...
...
clock_t start = clock();
...
...
clock_t end = clock();
printf("\nTime spent: %d\n",(end-start)/ CLOCKS_PER_SEC);
```

Outputting minimum cost and time spent in each method is enough. It is optional to output the order of matrix multiplication (refer to sample output), but it carries a bonus marks of 2 for the method in part II.

Submit a single .c or .cpp file. Your code should get compiled properly by gcc or g++ compiler.

Sample Output

```
palash@palash-ThinkPad-X1-Yoga-3rd:~$ ./a.out
Write number of matrices: 6
Write their dimensions (7 positive integers):
6 7 3 1 2 4 5
Minimum cost is 483
((A_1(A_2,A_3))((A_4,A_5)A_6))
Time spent by efficient method: 0 seconds.
Minimum cost is 483
Time spent by recursive method: 0 seconds.

palash@palash-ThinkPad-X1-Yoga-3rd:~$./a.out
Write number of matrices: 7
Write their dimensions (8 positive integers):
4 9 6 8 3 4 9 6
Minimum cost is 5202
((A_1(A_2(A_3,A_4)))((A_5,A_6)A_7))
Time spent by efficient method: 0 seconds.
Minimum cost is 5202
Time spent by recursive method: 0 seconds.

palash@palash-ThinkPad-X1-Yoga-3rd:~$ ./a.out
Write number of matrices: 21
Write their dimensions (22 positive integers):
6 7 3 1 2 4 5 9 4 2 8 3 1 2 4 5 9 2 5 3 6 7
Minimum cost is 2047
((A_1(A_2*A_3))((((((((((((((A_4*A_5)A_6)A_7)A_8)
A_9)(A_10(A_11*A_12)))A_13)A_14)A_15)A_16)A_17)A_18)A_19)A_20)A_21))
Time spent by efficient method: 0 seconds.
Minimum cost is 2047
Time spent by recursive method: 22 seconds.
```

Policy on Plagiarism

Academic integrity is expected from all the students. You should work on the assignment/exam consulting only the material we share with you. You are required to properly mention/cite anything else you look at. Any student submitting plagiarized code will be penalized heavily. Repeated violators of our policy will be deregistered from the course. Read this to know what is plagiarism.