

实验1.1：求矩阵链乘最优方案

文件架构

```
└─ ex1
  │  └─ input
  │      └─ 1_1_input.txt
  │  └─ output
  │      └─ result.txt
  │      └─ table_m.csv
  │      └─ table_s.csv
  │      └─ time.txt
  └─ src
      └─ IO.cpp
      └─ IO.h
      └─ algorithm.cpp
      └─ algorithm.h
      └─ file.h
      └─ main.cpp
      └─ main.exe
      └─ main.h
      └─ timing.cpp
      └─ timing.h
```

实验目标

求最优链乘方案及最少乘法运算次数，记录运行时间，仿照P214 图15-5，输出m表和s表。

实验工具

WSL, Vscode, Excel

实验结果与说明

input:

ouput:

result.txt:

第一行表示最少乘法运算次数，第二行表示最优链乘方案

154865959097238
(A(((AA)A)A))

table_m.csv:

P214 图15-5 m表

,5,4,3,2,1
1,154865959097238,128049683226820,74062781976714,15903764653528,0
2,138766801119366,105723424955724,43981152513978,0
3,183439291324068,119490227350806,0
4,120958281818244,0
5,0

table_s.csv:

P214 图15-5 s表

,5,4,3,2
1,1,1,1,1
2,4,3,2
3,4,3
4,4

time.txt:

运行时间

760

实验方法

动态规划

实验分析

代码解读

algorithm.cpp中 OPT Dynamic() 是主要算法，其按照

$$m[i,j] = \min_k \{m[i,k] + m[k+1,j] + p_i * p_k * p_j\}$$

$$s[i,j] = k$$

规则生成m表和s表。

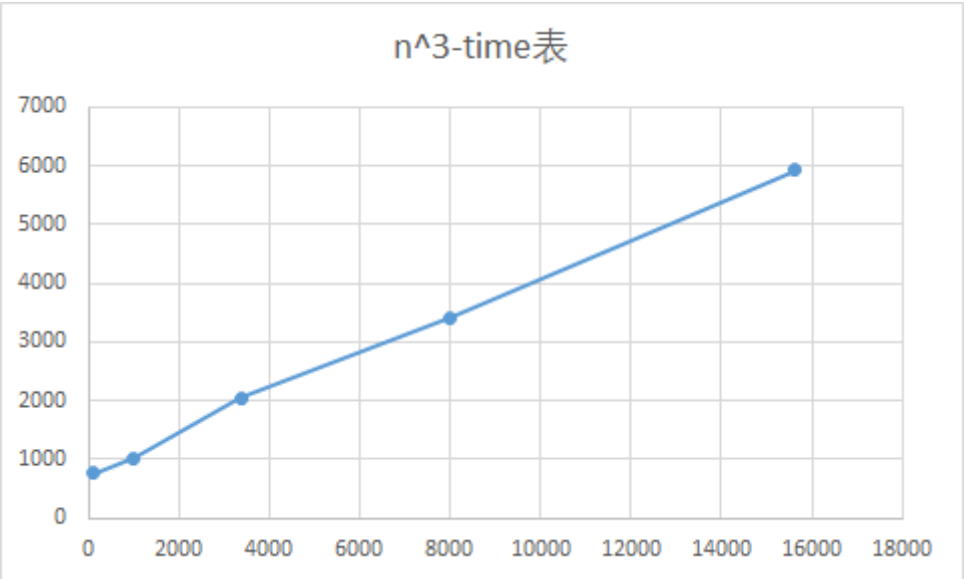
IO.cpp中输出按照P215 PRINT-OPTIMAL-PARENS算法输出最优链乘方案。

运行时间与复杂度分析

理论时间复杂度: $O(n^3)$

n	time	time/ n^3
5	760	6.08
10	1019	1.02
15	2050	0.61
20	3408	0.43
25	5930	0.38

曲线分析



横坐标为 n^3 ,纵坐标为 $time$,随着n的增大，越接近线性

在n较小时实际时间与理论时间相距较远，在n逐渐变大时，理论时间与实际时间比值趋于稳定。

m表:

		j				
		5	4	3	2	1
i	1	154865959097238	128049683226820	74062781976714	15903764653528	0
	2	138766801119366	105723424955724	43981152513978	0	
	3	183439291324068	119490227350806	0		
	4	120958281818244	0			
	5	0				

s表:

		j			
		5	4	3	2
i	1	1	1	1	1
	2	4	3	2	
	3	4	3		
	4	4			