

中国科学技术大学计算机学院

## 算法基础实验报告

### 实验二

### 动态规划算法

学    号：PB18071569

姓    名：张万林

专    业：计算机科学与技术

指导老师：顾乃杰

中国科学技术大学计算机学院

2020 年 12 月 09 日

## 一、实验目的

1. 掌握使用动态规划法解决问题。
2. 学会使用 FFT 方法。

## 二、实验内容

### 1. 求矩阵链乘最优方案

n 个矩阵链乘，求最优链乘方案，使链乘过程中乘法运算次数最少。

n 的取值 5, 10, 15, 20, 25，矩阵大小见 2\_1\_input.txt。

求最优链乘方案及最少乘法运算次数，记录运行时间，画出曲线分析。

仿照 P214 图 15-5，打印 n=5 时的结果并截图。

### 2. FFT。

多项式  $A(x) = \sum_{i=0}^{n-1} a_i x^i$ ，系数表示为  $(a_0, a_1, \dots, a_{n-1})$ 。

n 取  $2^3, 2^4, \dots, 2^8$ ，不同规模下的 A 见 2\_2\_input.txt。

用 FFT 求 A 在  $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$  处的值。

记录运行时间，画出曲线分析；打印  $n = 2^3$  时的结果并截图。

### 三、实验设备与环境

- 1、 硬件条件：一台 PC 机
- 2、 软件条件：VS code 编辑器 ， gcc 编译器 。

### 四、实验方法和步骤

1. 建立一个根文件夹实验需建立根文件夹，文件夹名称为:编号-姓名-学号-project2，在根文件夹下需包括实验报告和 ex1 子文件夹，和 ex2 文件夹。实验报告命名为编号-姓名-学号- project2.pdf， ex1 子文件夹和 ex2 子文件夹又包含 3 个子文件夹:

input 文件夹:存放输入数据

src 文件夹:源程序

output 文件夹:输出数据

2. 写矩阵链乘算法 ， 这一步先实现自底向上的动态规划问题求解函数 `long int Matrix_chain(int scale ,long int *p ,int path[MAX_SIZE][MAX_SIZE])` 。这个是按照书上算法，按部就搬， 然后实现递归的打印函数 。 最后在主函数里进行读文件，写文件 。

3. 写 FFT 算法 ， 这一步我用了蝴蝶操作，先写一个求二进制逆序数的函数 `void bit_reverse_copy ( )` ； 然后再自底向上的求 FFT 问题。最后在主函数里进行读文件，写文件 。

## 五、 实验结果与分析

1.先展示规模为 5 时的结果。

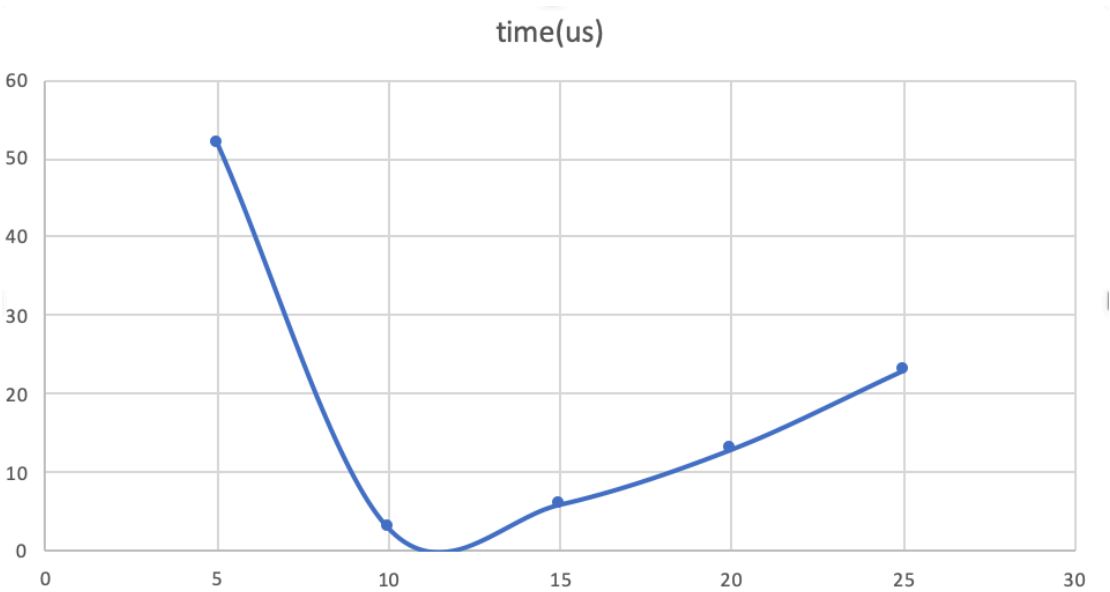
上面的矩阵是代价 ， 下面的矩阵是 path。

```
lin/workspace/CLRS/122-张万林-PB18071569-project2/ex1/src/"Matrix_Chain_Mult
=====规模为5时的结果 =====
0      15903764653528  74062781976714  128049683226820      154865959097238
0      0      43981152513978  105723424955724      138766801119366
0      0      0      119490227350806      183439291324068
0      0      0      0      120958281818244
0      0      0      0      0

0      0      0      0      0
0      0      1      2      3
0      0      0      2      3
0      0      0      0      3
0      0      0      0      0
ideven-Mac:src_zhangwanlin$
```

图 1.

2. 画出时间—规模 图，并分析 。



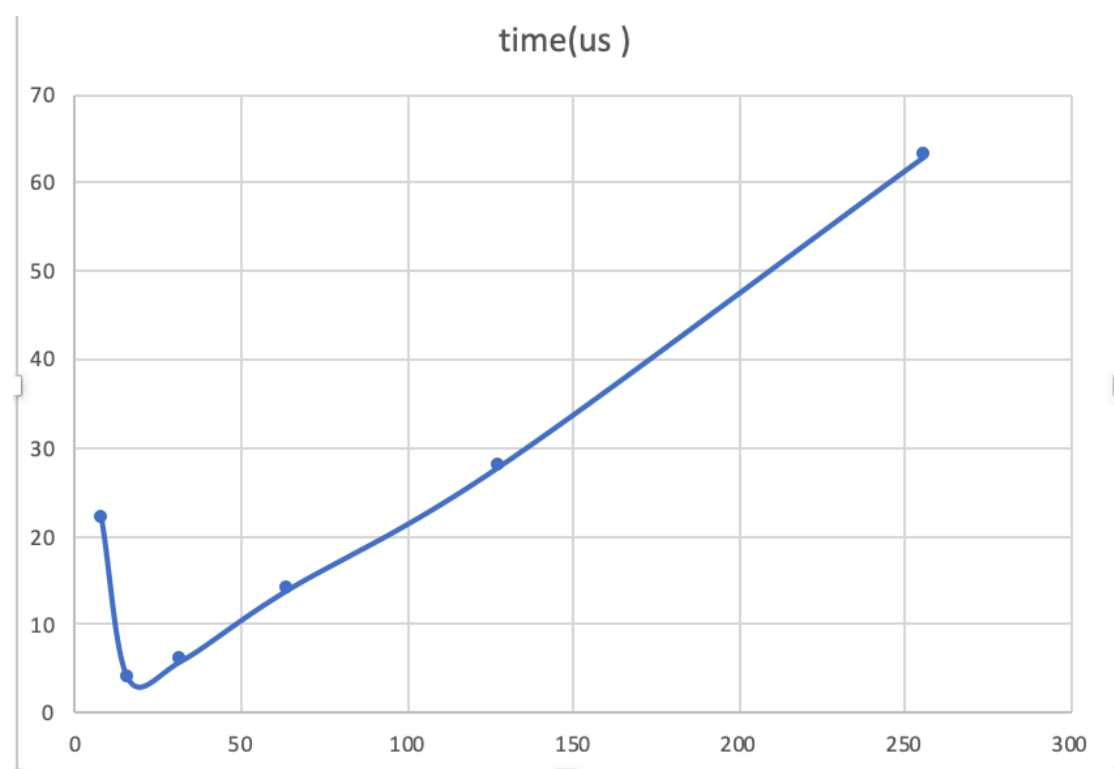
分析，除了  $n = 5$  时的数据不符合预期，其他的数据 符合的还是比较好的，分析其原因可能是因为，第一组数据在传参数是，会有 `cache miss` 等 或其他情况发生。而在后续的规模时是没有这种情况的，所以说第一组数据没有参考价值。为了验证这一观点，我又作了一个小的测试在下面介绍。

FFT 实验。

1.  $N = 8$  时的结果如图：

-10.000000    15.778175    5.000000    0.221825    -8.000000    0.221825    5.000000    15.778175

2. 分析时间曲线



分析，

FFT 与上一个问题一样，也是第一组数据不符合 预期。分析同上，

下面是我的验证：

我修改了输入，规模的次序，先输入大规模，得到，如下的截图结果。

```
The 256 scale problem costs 0.000092 s ...
The 128 scale problem costs 0.000031 s ...
The 64 scale problem costs 0.000014 s ...
The 32 scale problem costs 0.000006 s ...
The 16 scale problem costs 0.000003 s ...
```

可以看出 256 规模的时间 明显增常，而规模为 16 的代码时间成了预期的时间。可以看出这个意外状况与输入的数据是什么无关，而与函数调用的时机有关。