

西安电子科技大学

算法分析与设计（本科） 上机报告



学 院：_____ 软件学院

专 业：_____ 软件工程

方 向：_____ 云计算方向

姓 名：_____ 孙 晖

学 号：_____ 15130120141

目录

一、上次实验错题.....3

二、实验内容.....3

三、实验过程.....3

 3.1 实验一.....3

 3.1.1 实验内容.....3

 3.1.2 实验过程.....4

 3.1.3 实验结果.....4

 3.1.4 实验小结.....5

 3.2 实验二.....5

 3.2.1 实验内容.....5

 3.2.2 实验过程.....5

 3.2.3 实验结果.....5

 3.2.4 实验小结.....6

 3.3 实验三.....7

 3.3.1 实验内容.....7

 3.3.2 实验过程.....7

 3.3.3 实验结果.....7

 3.3.4 实验小结.....7

 3.4 实验四.....8

 3.4.1 实验内容.....8

 3.4.2 实验过程.....8

 3.4.3 实验结果.....8

 3.4.4 实验小结.....8

 3.5 实验五.....9

 3.5.1 实验内容.....9

 3.5.2 实验过程.....9

 3.5.3 实验结果.....9

 3.5.4 实验小结.....9

四、本次实验小结.....9

一、上次实验错题

首先，我想对上一次作业，也就是实验一的第五题的前半部分做一个解释，关于 n 个相等的数用快排需要比较多少次的这个问题，并不是我所想的那样是最优情况，实际上 n 个相等的数用快排需要的比较次数恰恰是最多的那种情况，因为比较条件是小于等于，意味着 n 个数相等需要最多的比较次数。因此实验一的第五题我写错了，在此进行更正。

N 个相等的数进行比较，比较次数为 $1+2+3+\dots+n-1=\frac{n(n-1)}{2}$ 次比较次数。

错误的原因是我思维上的误区。

二、实验内容

本次实验是关于动态规划问题，一共有五道题。

题目一：矩阵链乘的乘积问题，这一题要求我们给出一系列矩阵的最优链乘方法，通过加括号的方式给出最优的情况。题目给出了四个例子。

题目二：最长公共子序列问题。题目给出了两个例子。

题目三：最长公共子串问题，和第二题进行对比。

题目四：最大和问题，就是给定一系列整数，可正可负，找出任意长度的一个串使得这个串的所有数加在一起和最大。

题目五：最短路径问题。找到一个图中从 0 到 15 最短的走法。

三、实验过程

3.1 实验一

3.1.1 实验内容

矩阵链乘的乘积问题，这一题要求我们给出一系列矩阵的最优链乘方法，通过加括号的方式给出最优的情况。题目给出了四个例子。

- a) <3, 5, 2, 1, 10>
- b) <2, 7, 3, 6, 10>
- c) <10, 3, 15, 12, 7, 2>
- d) <7, 2, 4, 15, 20, 5>

3.1.2 实验过程

首先对<3,5,2,1,10>这样一个数组序列进行一下说明,这个数组序列实际上是5个矩阵,这五个矩阵的规模是 $3 \times 5, 5 \times 2, 2 \times 1, 1 \times 10$ 这个样子,之所以这样简写是因为,矩阵要想能够相乘,必定是第一个矩阵列的数量要和第二个矩阵行的数量进行乘积运算,因此可以这样简写。按照老师给的伪代码和讲解,可以看出,实际上只需要把那个倒三角形的表填出来就可以了。

3.1.3 实验结果

对 ac 两个个矩阵链进行运行得出的最终结果截图如下:

```
void main()
{
    int k, s[n][n];
    int p[5] = { 3, 5, 2, 1, 10 };
    k = matrixChainOrder(p, s);
    printf("%d个矩阵相乘所需的标量乘法的最小值为: %d\n", n - 1, k);
    printf("最终的最优全括号形式为: \n");
    printOptimalParens(s, 1, 4);
    getchar();
}
```

图 1 在 main 函数中给定好矩阵链 a

```
4个矩阵相乘所需的标量乘法的最小值为: 55
最终的最优全括号形式为:
((A1(A2A3))A4) _
```

图 2 矩阵链 a 的结果

```
void main()
{
    int k, s[n][n];
    int p[6] = { 10, 3, 15, 12, 7, 2 };
    k = matrixChainOrder(p, s);
    printf("%d个矩阵相乘所需的标量乘法的最小值为: %d\n", n - 1, k);
    printf("最终的最优全括号形式为: \n");
    printOptimalParens(s, 1, 5);
    getchar();
}
```

图 3 在 main 函数中给定好矩阵链 c

```
5个矩阵相乘所需的标量乘法的最小值为: 678
最终的最优全括号形式为:
(A1(A2(A3(A4A5)))) _
```

图 4 矩阵链 c 的结果

3.1.4 实验小结

算法是那种难者不会，会者不难的东西，没有用到编程语言中太难的地方，都是编程语言最简单的知识就可以做出来的题，而且老师还给有伪代码，我认为只要把思路搞懂，写出代码都是时间问题，即使有地方报错，改一改就行了，主要的还是思路和伪代码看懂。

3.2 实验二

3.2.1 实验内容

最长公共子序列问题。题目给出了两个例子。

1) X: xzyzzyx Y: zxyyzxz

2) X:MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCALLAAQANKESSSESFISRLLAIV
AD

Y:MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCTLLAAQANKENSNESFISRLLAIVAG

这道题是和第三题相对比的，公共子序列不要求子序列要连续，而公共子串要求这个子串是要连续的。

3.2.2 实验过程

思路：设 $A\{a_1, a_2, \dots, a_n\}$ $B\{b_1, b_2, \dots, b_m\}$

若 $a_n == b_m$; 则 $LCS\langle A, B \rangle = LCS\langle A(n-1), B(m-1) \rangle + a_n/b_m$

若 $a_n \neq b_m$; 则 $LCS\langle A, B \rangle = LCS\langle A(n-1), B \rangle$ 或 $LCS\langle A, B \rangle = LCS\langle A, B(m-1) \rangle$, 即 $LCS\langle A, B \rangle = \max\{LCS\langle A(n-1), B \rangle, LCS\langle A, B(m-1) \rangle\}$

在网上看到有人将老师给的伪代码完全照搬着去写代码，比如 LCS 的矩阵中需要箭头来指示子序列的走向，代码中就用 up、left、这些单词来表示，然后又用 switch 语句把这些 up、left 情况纷纷对应起来，感觉完全没有必要，只需要一开始就用数组的下标把 up、left 在逻辑上体现出来就行，虽然网上的代码更容易理解，是伪代码的百分百实现，但是感觉有些多次一举，而其中将 LCS 打印出来的代码感觉很有意思，很形象，就抄了下来。

3.2.3 实验结果

```

void main()
{
    char X[100] = {'x','z','y','z','z','y','x'};
    char Y[100] = {'z','x','y','y','z','x','z'};
    int i, j, s;

    printf("LCS的矩阵如下: \n");
    s = LCS(X, Y);
    printf("X和Y的LCS: %d \n", s);
    getchar();
}

```

图 1 在 main 函数中给定好两个序列

```

LCS的矩阵如下:
  z  x  y  y  z  x  z
x  0  1  1  1  1  1  1
z  1  1  1  1  2  2  2
y  1  1  2  2  2  2  2
z  1  1  2  2  3  3  3
z  1  1  2  2  3  3  4
y  1  1  2  3  3  3  4
x  1  2  2  3  3  4  4
X和Y的LCS是:zyyx
X和Y的LCS: 4

```

图 2 输出结果如图所示

3.2.4 实验小结

老师在上课时指出考试中会考到 LCS 矩阵这个东西，也就是

j	0	1	2	3	4	5	6
i	y_i	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0
1	A	0	0	0	0	1	1
2	B	0	1	1	1	2	2
3	C	0	1	1	2	2	2
4	B	0	1	1	2	3	3
5	D	0	1	2	2	3	3
6	A	0	1	2	3	3	4
7	B	0	1	2	3	4	4

主要记住算法中何时左上，何时上，何时左就可以理解 LCS 算法的本质了。也就是 PRINT-LCS 这个函数。

3.3 实验三

3.3.1 实验内容

公共子串问题，和题目二相对应的，要求是子串必须是连续的。

3.3.2 实验过程

一开始想着相对于公共子序列而言，公共子串要简单很多，无非是两个序列，把第一个序列从头开始和第二个序列比下去，遇到相同的，第一个序列再后移一位接着和第二个序列比下去，也就是网上人们说的暴力搜索。后来发现，别人有更好的方法，但是涉及到很多指针我没有看懂，这里暴露了指针问题的不足与盲点，闲暇时可以继续深究下去。

3.3.3 实验结果

```
void main() {  
    char str1[50] = { 'x', 'z', 'y', 'z', 'z', 'y', 'x' };  
    char str2[50] = { 'z', 'x', 'y', 'y', 'z', 'x', 'z' };  
    char str3[50];  
  
    getCommon(str1, str2, str3);  
    printf("%s\n", str3);  
    getchar();  
}
```

图 1 在头文件中初始化两个字符串序列

A screenshot of a terminal window with a black background. The text 'xz' is displayed in a light blue or cyan color. The text is positioned on the left side of the terminal, with a vertical cursor line visible to its left.

图 2 输出结果为 xz

3.3.4 实验小结

LCS_Substring 暴露了我在学习指针问题中的不足，可能当时是会的，现在又开始迷迷糊糊了。要总结一下。

3.4 实验四

3.4.1 实验内容

最大和问题：老师给的例子是算（-2,11,-4,13,-5,-2）。结合老师上课时讲的内容，老师希望的答案是，并且给出最大值是多少。

3.4.2 实验过程

在写这道题的时候，我并没有完全按照老师给的 DP 算法去写，而是这样思考的，如果这道题全是负数的情况，那么就意味着什么都不选是最好的情况，答案为零。那么我把答案是负数的情况重置为零就好。在比较一下当前的值和加上下一个数的值的大小就可以得出结论。

3.4.3 实验结果

```
int main() {  
    int a[100] = {-2, 11, -4, 13, -5, -2};  
    DPMax_Sum(a, 6);  
    getchar();  
    return 0;  
}
```

图 1 规定好这个序列以及输入的数字数量 6

最大值为：20

图 2 输出的结果为 20

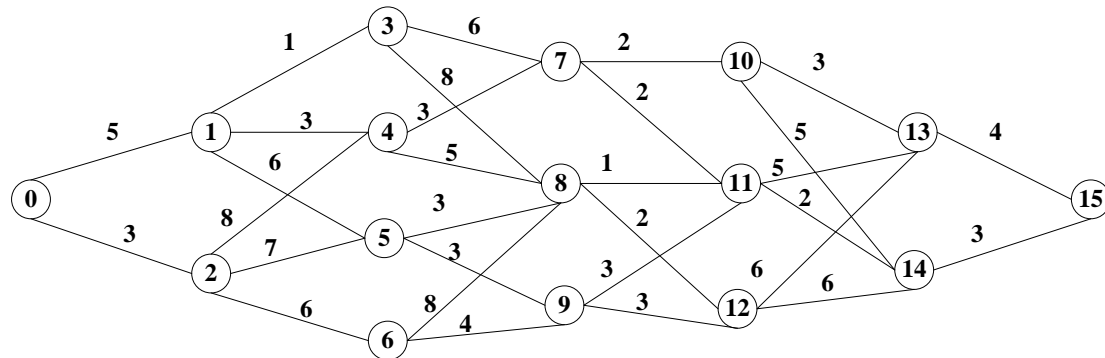
3.4.4 实验小结

我写的这个程序只能特定的解在 main 函数里规定好的情况，因为参数 n 代表了输入数组中的数字个数，如果不这样就需要再写一个函数得出数组的长度，或者用 C 语言的库函数给出数组长度，这里我没有给出究竟是从数组哪一个到哪一个的结果，倘若需要给出，我就需要重写整个函数了。

3.5 实验五

3.5.1 实验内容

找到多段图的最短路径，要求这个最短路径从节点 0 到节点 15.



3.5.2 实验过程

一开始看到这一题的时候以为是 dijkstra 算法，后来才明白是多段图的最短路径。在学习动态规划这一章的时候，总感觉学的很快就过去了，只是对几张重要的图有印象，因此百度了多段图的最短路径，学会了解法。

3.5.3 实验结果

由多段图的性质可以知道，只需要使得每一段的路径保持最小就能使所有从开始到结尾的结点路径保持最小。

```
0 --> 1 --> 4 --> 7 --> 11 --> 14 --> 15
```

图 1 最短路径如图所示

最短路径长度为 18.

3.5.4 实验小结

关于这种方法，碰到路径相同时，算法应该怎么写呢？

四、本次实验小结

- [1] 动态规划老师上课时讲了只能用于最优子结构问题，即局部最优解能够决定全局最优解。对于一个问题如果能够分成子问题来解决，那么可以用，就像老师上课讲的零件调配问题。

- [2] 动态规划算法分为四个步骤:
- [3] 描述最优解的结构
- [4] 递归定义最优解的值
- [5] 按自底向上的方式计算最优解的值
- [6] 由计算出的结果构造一个最优解。
- [7] 关于用 VS 会出现 scanf 要用 scanf_s 的处理方法。
- [8] <https://jingyan.baidu.com/article/363872ecd87e5d6e4ba16f1c.html>
- [9] <>引用的是编译器的类库路径里面的头文件" "引用的是你程序目录的相对路径中的头文件。
- [10] 若排版出错建议看 pdf 版的实验报告。
- [11] 我个人始终认为算法是一门很有意思的课程，而且对于以后找工作也很有用，因此要认真学习。