目录

一、实验题目

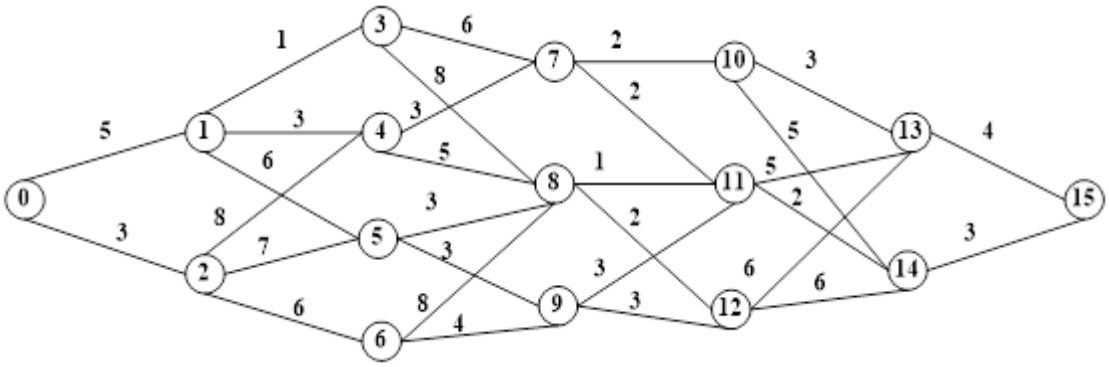
1、Matrix-chain product.

2、Longest Common Subsequence (LCS).

3、Longest Common Substring.

4、Max Sum

5、Shortest path in multistage graphs. Find the shortest path from 0 to 15 for the following graph.A multistage graph is a graph (1) G=(V,E) with V partitioned into K >= 2 disjoint subsets such that if (a,b) is in E, then a is in Vi , and b is in Vi+1 for some subsets in the partition; and (2) | V1 | = | VK | = 1.



二、实验目的

了解动态规划法的思想，使用的题目类型，掌握使用动态规划法分析解决问题。

三、实验设计与分析

1、Matrix-chain product.

首先，矩阵链乘不需要计算结果，只是找到计算次数最小的乘法顺序，可以

将问题转化为求解矩阵链乘需要的最小乘法次数。

其次，如果使用暴力求解的方法时间复杂度过大，因为问题可以分解为更小的问题，可以考虑使用动态规划法求解。该问题的基本思路是找出每个子序列的最优解再将子序列的最优解合并找到矩阵链乘的最优解，合并的过程为子序列最优解相加再加上子序列之间相乘的成本；对每个可以用括号进行分割的点执行上述操作，找到矩阵链乘的最优解，递推表达式如下：

$$m[i,j]=\begin{cases}0 & \text{if } i=j\\ \min\limits_{i\le k<j}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\} & \text{if } i<j\end{cases}$$

- $m[i,k]$ = optimal cost for $A_i\times\ldots\times A_k$
- $m[k+1,j]$ = optimal cost for $A_{k+1}\times\ldots\times A_j$
- $p_{i-1}p_kp_j$ = cost for $(A_i\times\ldots\times A_k)\times(A_{k+1}\times\ldots\times A_j)$

具体算法思路为三层循环，最外层循环从 2 个矩阵相乘递增到 n 个矩阵相乘；第二层循环从第一个矩阵开始计算，遍历相应个数的矩阵相乘的代价；最内层循环寻找并记录最优解。

2、Longest Common Subsequence (LCS).

由最长公共子序列的性质可知，对于长度为 m 的 A 序列和长度为 n 的 B 序列，只需要求得 A[1,m-1]与 B[n-1]的最长公共子序列，如果 A[m]个元素与 B[n]个相同则在上述结果加上 A 序列的第 m 个元素，否则将取 A[m]和 B[n-1]的最长公共子序列和 A[m-1]和 B[n]的最长公共子序列的最大值。递推公式表达如下：

$$c[i,j]=\begin{cases}c[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } x_i=y_j\\ \max(c[i,j-1],c[i-1,j]) & \text{otherwise}\end{cases}$$

具体的算法设计是两层循环，分别从小到大对序列 A 和序列 B 遍历。

3、Longest Common Substring.

此题目与第二题的不同之处在于此题要求最长公共字串必须是连续的，遍历到两元素相同时与第二题相同，遍历到两元素不同时不需要记录，同时需要记录

当前最长子串长度。递推公式如下：

当A[i] != B[j]，dp[i][j] = 0

当A[i] == B[j]，

若i = 0 || j == 0，dp[i][j] = 1

否则 dp[i][j] = dp[i - 1][j - 1] + 1

具体的算法设计与第二题类似，只不过当两元素不同时将 dp[i][j]置 0，增加 max 记录当前最长公共子串长度

4、Max Sum

如果之前的序列和为整数，则对当前的数有增益，可以直接相加；如果之前的序列和为负数，则抛弃之前的序列和。使用 ans 记录当前的最大序列和。

5、Shortest path in multistage graphs.

给定的图具有一定的特点，同层之间没有连接，0 和 15 之间共有 5 层，所以当前节点到终点的最短路径就是当前节点到下一层可达节点的最短路径加上下一层可达节点到终点的最短路径。

四、实验环境

Windows 10、IntelliJ IDEA 2020.3.2 x64、jdk1.8.0_281

五、项目测试（功能和性能）

1、Matrix-chain product.

实验结果如下图所示

```
<3, 5, 2, 1,10>
最小代价为：55
((A1(A2A3))A4)
<2, 7, 3, 6, 10>
最小代价为：198
(((A1A2)A3)A4)
<10, 3, 15, 12, 7, 2>
最小代价为：678
(A1(A2(A3(A4A5))))
<7, 2, 4, 15, 20, 5>
最小代价为：990
(A1(((A2A3)A4)A5))
```

算法时间复杂度为 $O(n^3)$，使用括号表示最优解的矩阵乘积顺序，经验证结果正确

2、Longest Common Subsequence (LCS).

实验结果如下图所示

```
序列A为：xzyzzyx
序列B为：zxyyzxz
最长公共子序列长度为：4
序列A为：MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCALLAAQANKESSSESFISRLLAIVAD
序列B为：MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCTLLAAQANKENSNESFISRLLAIVAG
最长公共子序列长度为：56
```

算法时间复杂度为 O(mn)，m 和 n 分别代表两个序列的长度，经验证结果正确。

3、Longest Common Substring.

实验结果如下图所示

```
序列A为：xzyzzyx
序列B为：zxyyzxz
最长公共子串长度为：2
序列A为：MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCALLAAQANKESSSESFISRLLAIVAD
序列B为：MAEEEVAKLEKHLMLLRQEYVKLQKKLAETEKRCTLLAAQANKENSNESFISRLLAIVAG
最长公共子串长度为：34
```

算法时间复杂度同第二题，经验证结果正确

4、Max Sum

实验结果如下图

```
序列为：[-2, 11, -4, 13, -5, -2]
最大字序和为：20
```

算法时间复杂度为O(n)，经验证结果正确

5、Shortest path in multistage graphs.

实验结果如下图

```
输入邻接矩阵为：
[0, 5, 3, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 0, 1000, 1, 3, 6, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 0, 1000, 8, 7, 6, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 0, 1000, 1000, 1000, 6, 8, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 0, 1000, 1000, 3, 5, 1000, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 0, 1000, 1000, 3, 3, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 8, 4, 1000, 1000, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 1000, 2, 2, 1000, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 1000, 1, 2, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 3, 3, 1000, 1000, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 1000, 3, 5, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 5, 2, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 6, 6, 1000]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 1000, 4]
[1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 3]
最短路径为：0->1->4->7->11->14->15
最短路径长度为18
```

时间复杂度为$O(n^2)$,经验证结果正确