北京郵電大學

Beijing University of Posts and Telecommunications

《算法分析—第三章实验》

实验报告

姓　　名　袁洁　胡敏臻

学　　号　2019211426　2019211424

班　　级　2019211307　2019211307

专　　业　计算机科学与技术

1 实验内容

1.1.1 最长公共子序列

利用“附件 1.最长公共子序列输入数据-2021”中给出的字符串 A, B, C, D, 分别找出下列两两字符串间的最长公共子串，并输出结果：

A-B,

C-D,

A-D,

C-B

1.1.2 最长递减子序列

利用最长公共子序列求解下列最长递减子序列问题：

给定由 n 个整数$a_1, a_2, \dots, a_n$构成的序列，在这个序列中随意删除一些元素后可得到一个子序列$a_i, a_j, a_k, \dots, a_m$，其中 1≤i≤m≤n，并且$a_i \geq a_j \geq a_k \geq \cdots \geq a_m$，则称序列$a_i, a_j, a_k, \dots, a_m$为原序列的一个递减子序列，长度最长的递减子序列即为原序列的最长递减子序列。

要求：

利用“附件 2.最大子段和输入数据-序列 1”、“附件 2.最大子段和输入数据-序列 2”，求这两个序列中的最长递减子序列。

1.2 最大字段和

针对“附件 2.最大子段和输入数据-序列 1”和“附件 2.最大子段和输入数据-序列 2”中给出的序列 1、序列 2, 分别计算其最大子段和。

要求：

1. 指出最大子段和在原序列中的位置
2. 给出最大子段和具体值

1.3 凸多边形最优三角剖分

算法一：利用：教科书上 O(n^3)算法，“附件 3-1.21 个基站凸多边形数据”和“附件 3-2.29 个基站凸多边形数据”，给出 21 凸多边形顶点数据、 29 凸多边形顶点数据，以顶点间的地理距离作为连接 2 个顶点的边、弦到的权值，对这 2 个凸多边形进行最优三角剖分
采用近似的复杂度为 O(n^2)的剖分算法，实现三角剖分。

算法二：对子问题{v(i-1),v(i),⋯,v(k),⋯v(j)}，从 K(<=3)个固定位置选断点 v(k)，比较这 K 个断点的目标值，从中选取最优断点。

要求：

1. 做图表示最优、次优三角剖分结果，可以手绘
2. 计算最优、次优三角剖分对应的目标值——边长弦长总和

1.4 0-1 背包问题

利用“附件 4.背包问题输入数据”给出的 2 组背包数据（背包容量、物品重量、物品价值），计算最优物品装载方案

要求：

给出最优方案中，

1. 各个物品是否被放入
2. 物品放入后的背包总重量、总价值

2 最长公共子序列

2.1 设计思路

最长公共子序列设计思路：根据最优子结构性质，建立子问题最优值的递归关系，其中c[i][j]：序列 X(i)和 Y(j)的最长公共子序列的长度，

1．边界条件：X(i)={x1,x2,……,xi}，Y(j)={y1,y2,……,yj}，当 i=0 或 j=0 时，即其中 1 个序列为空，则空序列是 X i 和 Yj 的最长公共子序列，故此时 C[i][j]=0;

2．其他情况下，由最优子结构性质可建立递归关系如下：

$$c[i][j]=\begin{cases}0 & i=0\ ||\ j=0\\ c[i-1][j-1]+1 & i,j>0;x_i=y_j\\ \max\{c[i][j-1],c[i-1][j]\} & i,j>0;x_i\neq y_j\end{cases}$$

构造最长公共子序列设计思路：

建立数组 b: b[i][j]记录 c[i][j]的值是由哪个子问题得到的(3 种情况之一)，用于后续构造最长公共子序列。

从 b[m,n]开始，依其值在数组 b 中搜索，会出现三种情况：

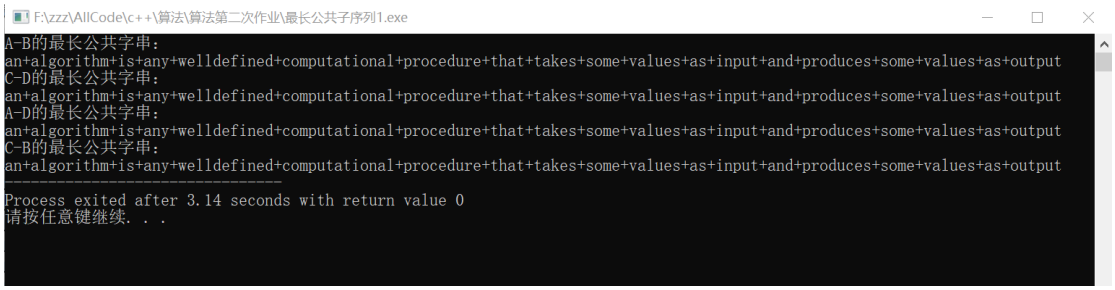
1．b[i,j]=1，Xi 和 Yj 的最长公共子序列由 Xi-1 和 Yj-1 的最长公共子序列，在尾部加上 Xi 得到

2．b[i,j]=2，Xi 和 Yj 的最长公共子序列与 Xi-1 和 Yj 的最长公共子序列相同

3．b[i,j]=3，Xi 和 Yj 的最长公共子序列与 Xi 和 Yj-1 的最长公共子序列相同

2.2 实验结果

我们需要分别找出下列两两字符串间的最长公共子串，输出结果：A-B, C-D, A-D, C-B



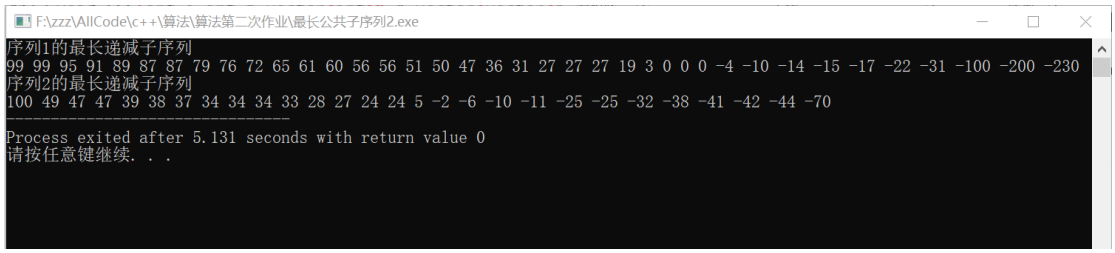
两两字符串的最长公共子串都是：

an+algorithm+is+any+welldefined+computational+procedure+that+takes+some+values+as+input+and+produces+some+values+as+output

2.3 最长递减子序列

在这题中，我们还需要求出两个序列的最长递减序列，其实这题的方法与求最长公共子序列类似，只是这时我们需要求的最长公共子序列为序列本身和其本身递减序列的最长公共子序列。

2.4 最长递减子序列结果



3 最大子段和

3.1 设计思路

根据分治法，分析左右子段的 s1、s2 与原序列最优值间的关系——需要知道段与子段之间最优值间的递归关系。

记$b[j]=\max\limits_{1\le i\le j}\{\sum_{k=i}^{j}a_k\}$，那么对所求最大字段和就有如下关系：

$$\max_{1\le i\le j\le n}\sum_{k=i}^{j}a_k=\max_{1\le j\le n}\max_{1\le i\le j}\sum_{k=i}^{j}a_k=\max_{1\le j\le n}b[j]$$

因此在设计之后，我们可以得到如下递归方程

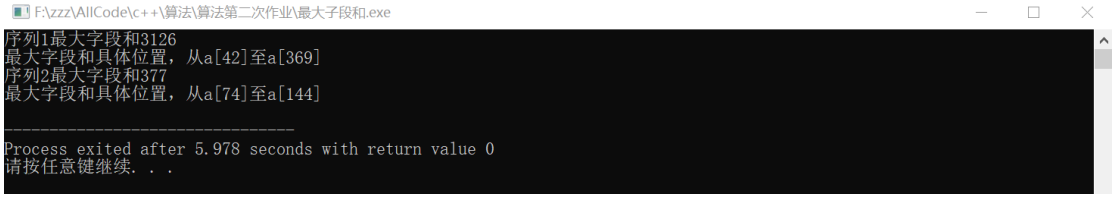
$$b[j]=\max\{b[j-1]+a[j],a[j]\}\quad 1\le j\le n$$

根据该递归方程我们可以求出最大字段和。

3.2 实验结果

我们需要求出序列 1 和序列 2：

1. 指出最大子段和在原序列中的位置
2. 给出最大子段和具体值



4 凸多边形最优三角剖分

4.1 实验内容

给定凸多边形 P，以及定义在由多边形的边和弦组成的三角形上的权函数 w(如欧氏距离)，要求:确定该凸多边形的三角剖分，使得即该三角剖分中诸三角形上权之和为最小——最优值。

4.2 设计思路

矩阵连乘/表达式的完全加括号方式问题与三角剖分问题具有相似性，矩阵连乘表达式的完全加括号问题可以表示为一棵完全二叉树，称为表达式的语法树，矩阵连乘表达式的完全加括号问题也可以表示为一棵完全二叉树， 称为表达式的语法树。树根节点对应 V0Vn，叶节点对应边 Vi-1Vi，非叶结点对应剖分多边形的弦边。

凸多边形三角剖分具有最优子结构性质，所以根据剖分顶点，可以将原问题分解为子问题，最后自底向上求解。求出递归表达式：

$$t[i][j]=\begin{cases}0 & i=j\\ \min\limits_{i\le k<j}\{t[i][k]+t[k+1][j]+w(v_{i-1}v_kv_j)\} & i<j\end{cases}$$

算法 1:断点 K 的选取采用**枚举方法**，对每个可能的顶点进行计算求解得出最小值。

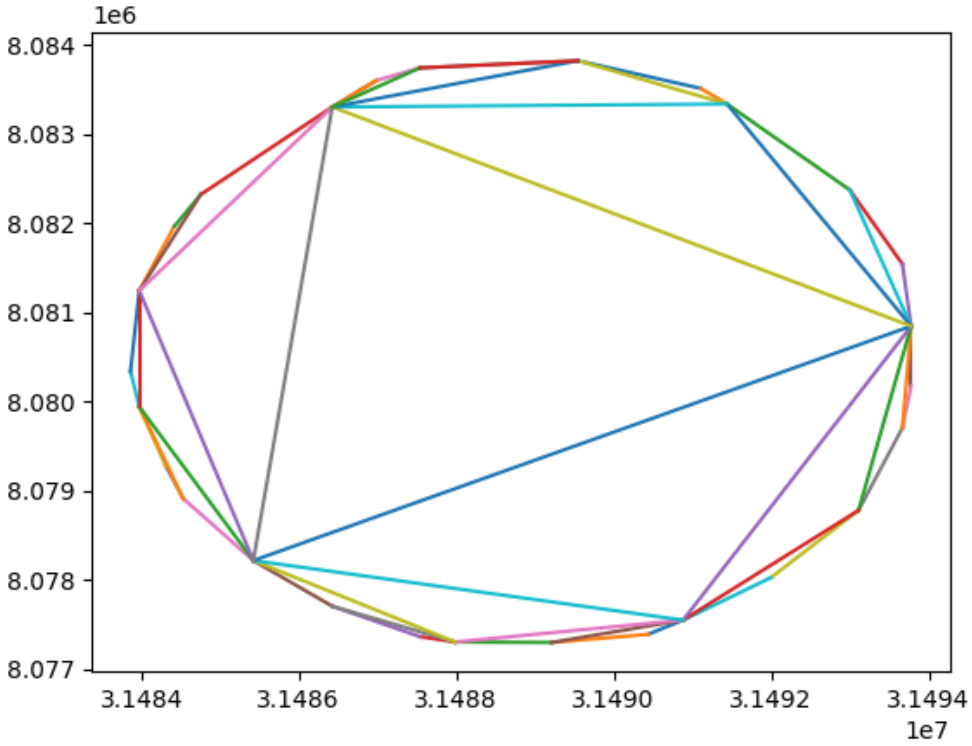
算法 2:断点 K 的选取从**固定位置进行选取**，比较这 K 个断点的目标值，从中选取最优

断点。我们组选用了 i+1，j-1，i+j/2 这三个位置作为断点。

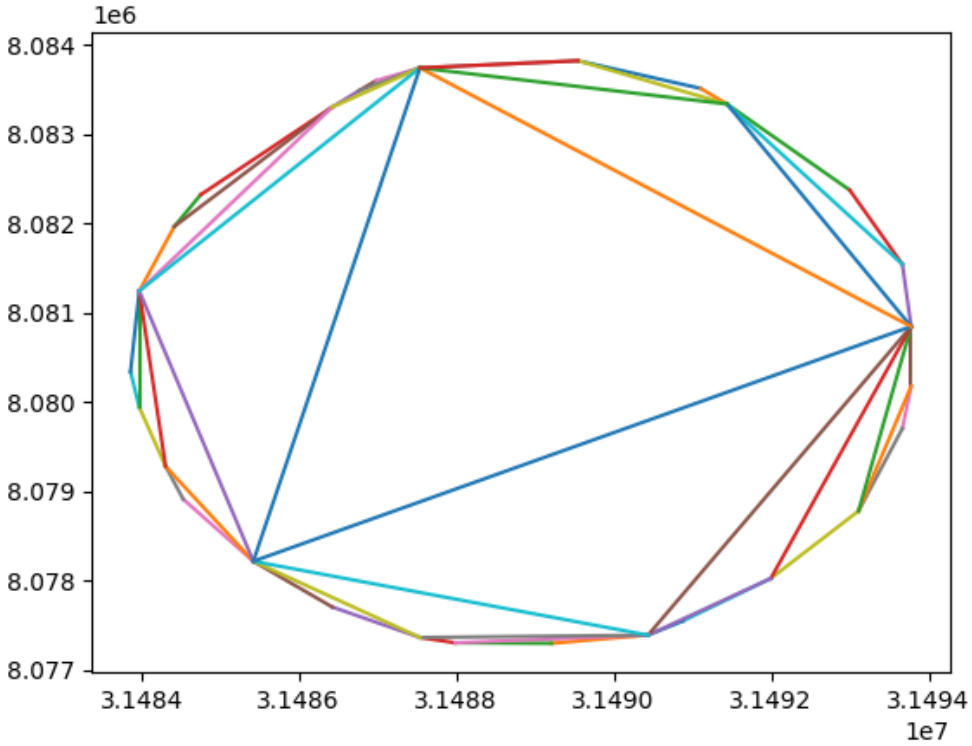
4.3 实验结果

4.3.1 算法一实验结果

29 个基站最优划分



29 个基站次优划分

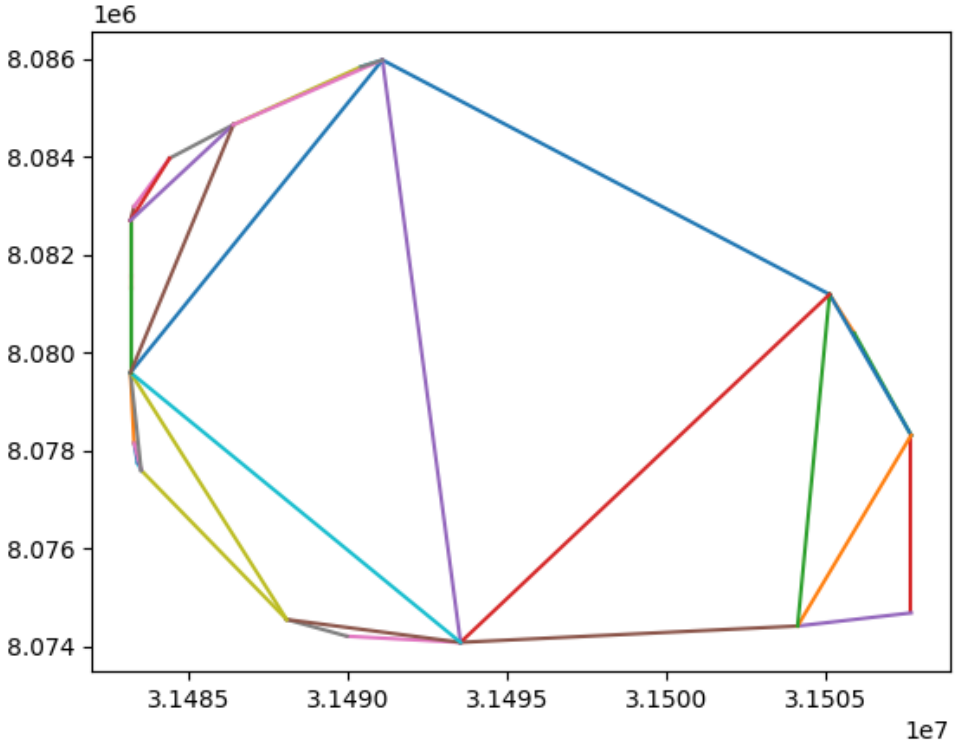


29 个基站最优划分——次优划分

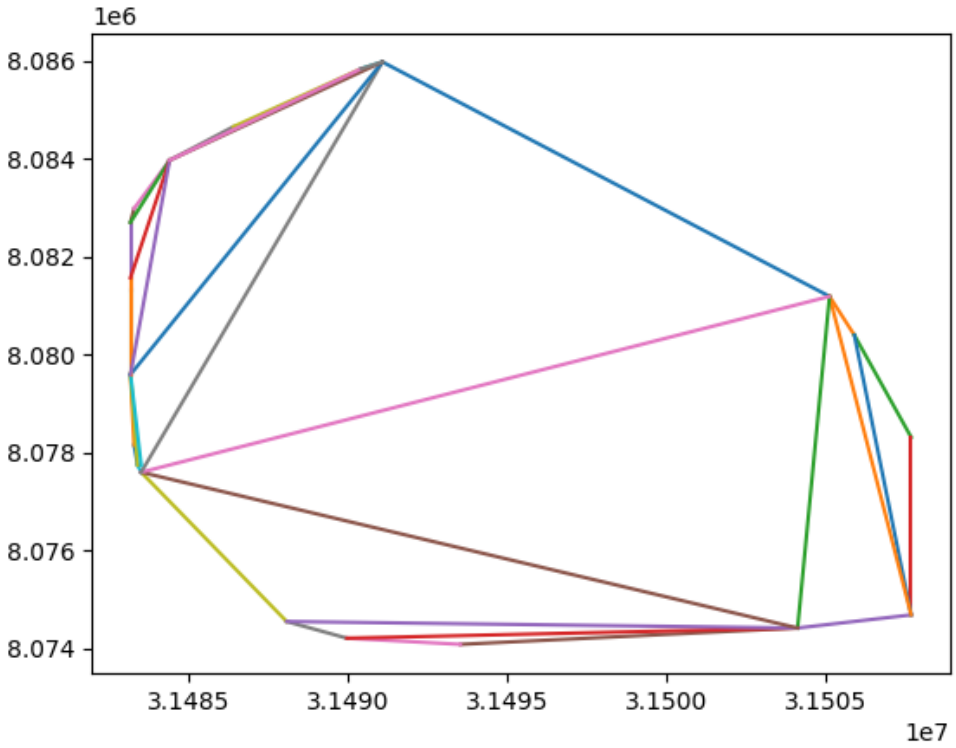
109809.246308 110055.274489

最优剖分：V0,V2:V1	次优剖分：V0,V2:V1
最优剖分：V3,V5:V4	次优剖分：V2,V4:V3
最优剖分：V2,V5:V3	次优剖分：V2,V5:V4
最优剖分：V5,V7:V6	次优剖分：V6,V8:V7
最优剖分：V5,V8:V7	次优剖分：V5,V8:V6
最优剖分：V8,V10:V9	次优剖分：V5,V9:V8
最优剖分：V5,V10:V8	次优剖分：V9,V11:V10
最优剖分：V10,V12:V11	次优剖分：V5,V11:V9
最优剖分：V10,V13:V12	次优剖分：V11,V13:V12
最优剖分：V13,V15:V14	次优剖分：V11,V14:V13
最优剖分：V13,V16:V15	次优剖分：V14,V16:V15
最优剖分：V10,V16:V13	次优剖分：V11,V16:V14
最优剖分：V5,V16:V10	次优剖分：V5,V16:V11
最优剖分：V17,V19:V18	次优剖分：V16,V18:V17
最优剖分：V16,V19:V17	次优剖分：V19,V21:V20
最优剖分：V19,V21:V20	次优剖分：V18,V21:V19
最优剖分：V16,V21:V19	次优剖分：V16,V21:V18
最优剖分：V21,V23:V22	次优剖分：V22,V24:V23
最优剖分：V21,V24:V23	次优剖分：V21,V24:V22
最优剖分：V16,V24:V21	次优剖分：V25,V27:V26
最优剖分：V5,V24:V16	次优剖分：V24,V27:V25
最优剖分：V2,V24:V5	次优剖分：V21,V27:V24
最优剖分：V0,V24:V2	次优剖分：V16,V27:V21
最优剖分：V24,V26:V25	次优剖分：V5,V27:V16
最优剖分：V24,V27:V26	次优剖分：V2,V27:V5
最优剖分：V0,V27:V24	次优剖分：V0,V27:V2

21 个基站最优划分



21 个基站次优划分

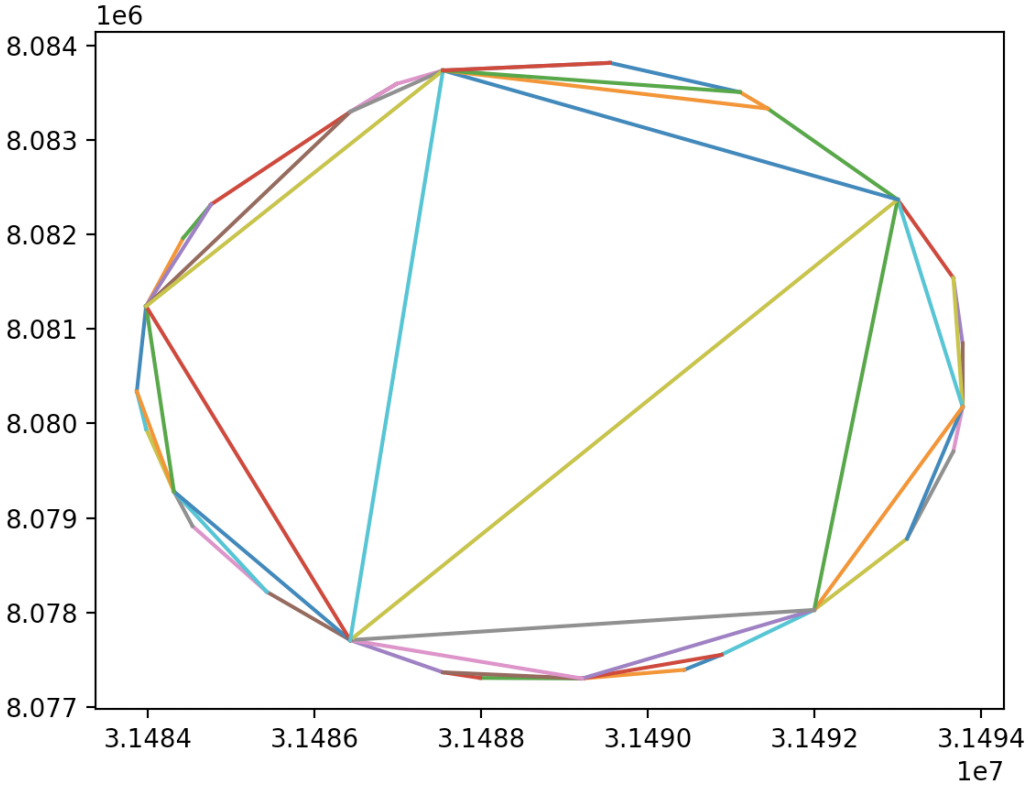


21 个基站最优划分——次优划分

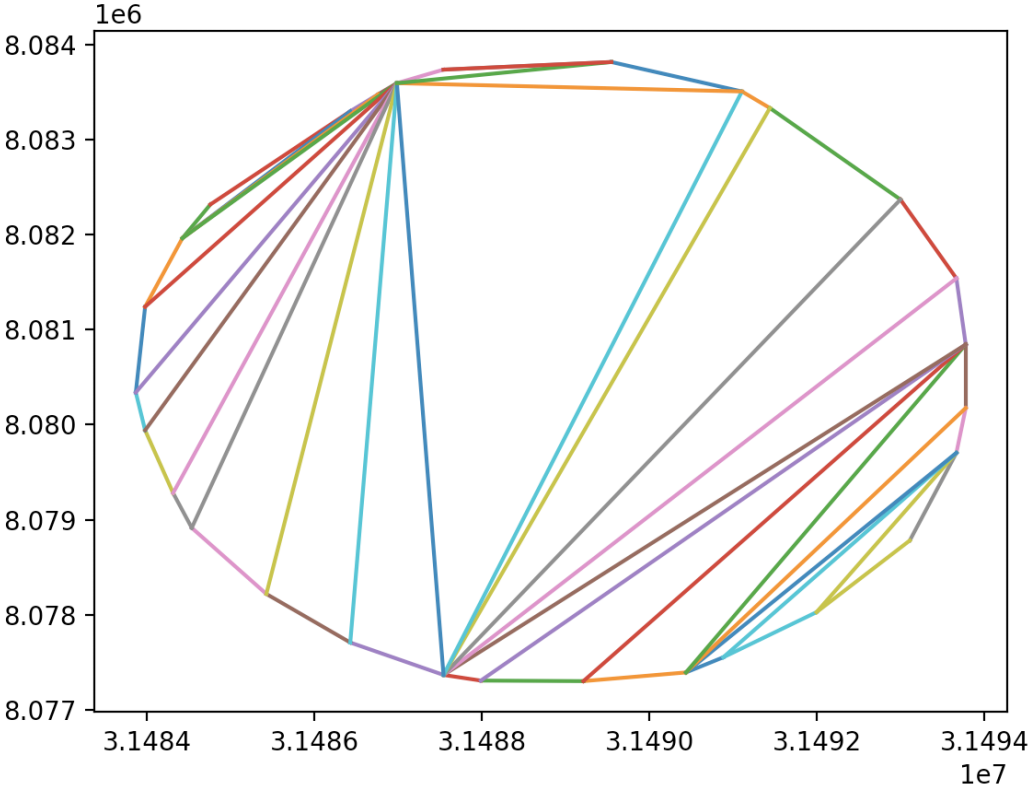
179002.987207	181080.646687
最优剖分: V1,V3:V2	次优剖分: V2,V4:V3
最优剖分: V3,V5:V4	次优剖分: V1,V4:V2
最优剖分: V1,V5:V3	次优剖分: V1,V5:V4
最优剖分: V1,V6:V5	次优剖分: V5,V7:V6
最优剖分: V0,V6:V1	次优剖分: V5,V8:V7
最优剖分: V6,V8:V7	次优剖分: V5,V9:V8
最优剖分: V9,V11:V10	次优剖分: V1,V9:V5
最优剖分: V9,V12:V11	次优剖分: V0,V9:V1
最优剖分: V8,V12:V9	次优剖分: V10,V12:V11
最优剖分: V6,V12:V8	次优剖分: V9,V12:V10
最优剖分: V0,V12:V6	次优剖分: V0,V12:V9
最优剖分: V13,V15:V14	次优剖分: V12,V14:V13
最优剖分: V12,V15:V13	次优剖分: V15,V17:V16
最优剖分: V15,V17:V16	次优剖分: V14,V17:V15
最优剖分: V15,V18:V17	次优剖分: V12,V17:V14
最优剖分: V12,V18:V15	次优剖分: V0,V17:V12
最优剖分: V0,V18:V12	次优剖分: V17,V19:V18
最优剖分: V0,V19:V18	次优剖分: V0,V19:V17

4.3.2 算法二实验结果

29 个基站最优划分


1e6
8.084
8.083
8.082
8.081
8.080
8.079
8.078
8.077
3.1484
3.1486
3.1488
3.1490
3.1492
3.1494
1e7

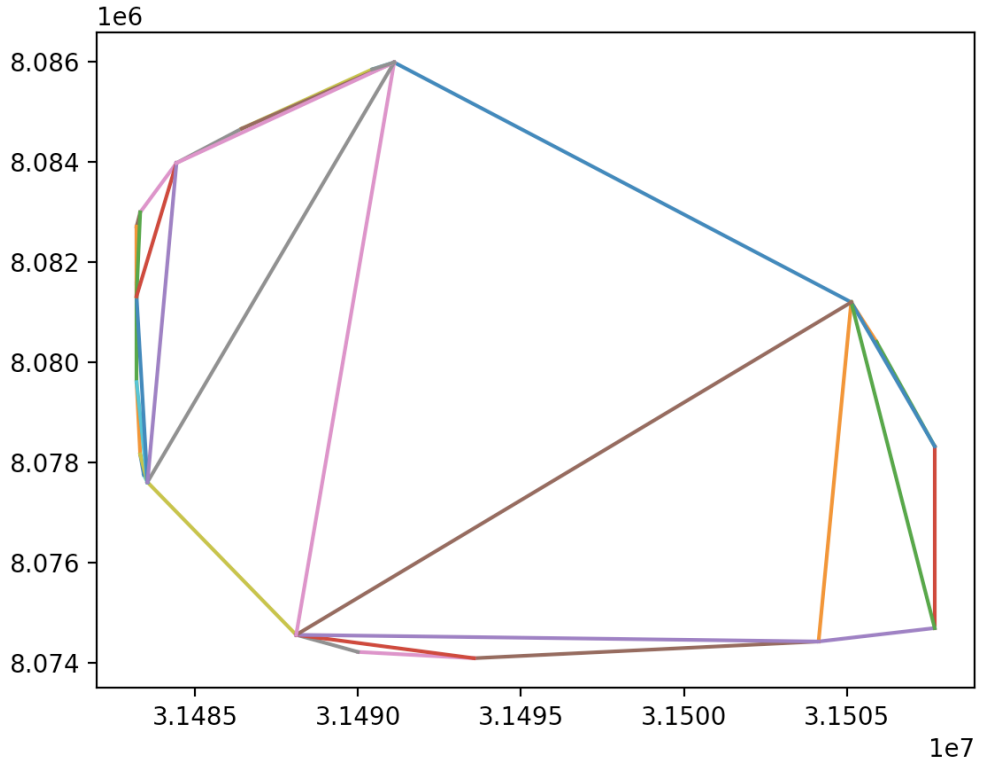
29 个基站次优划分


1e6
8.084
8.083
8.082
8.081
8.080
8.079
8.078
8.077
3.1484
3.1486
3.1488
3.1490
3.1492
3.1494
1e7

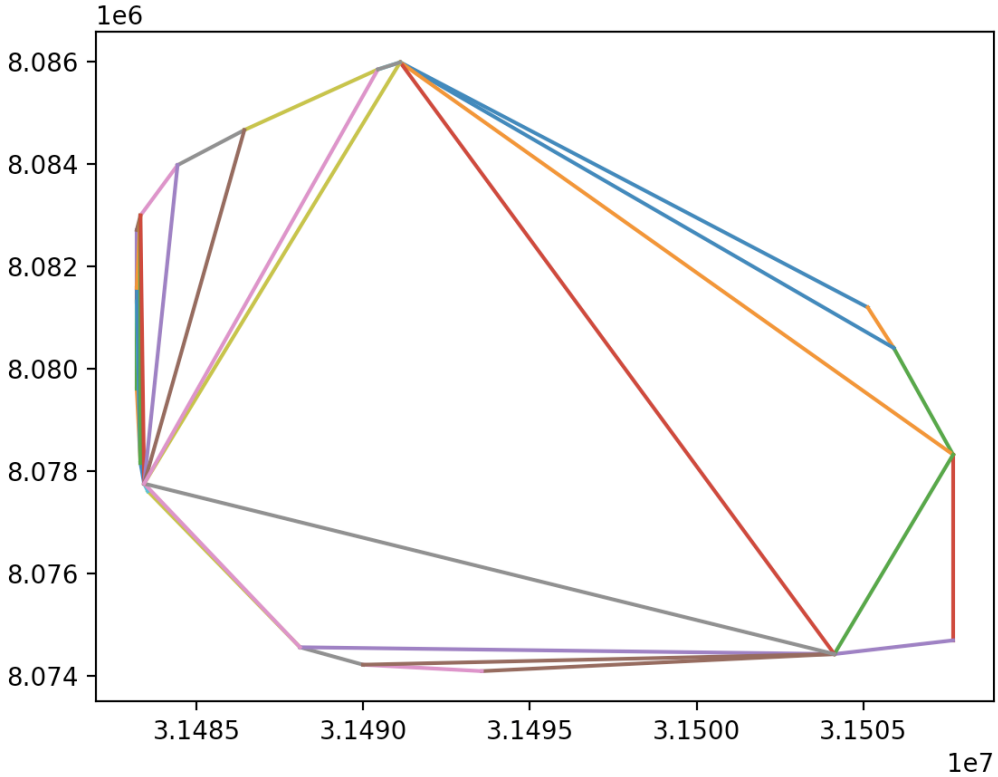
29 个基站最优划分——次优划分

113298.077943 114070.283683	
最优剖分: V4,V6:V5	次优剖分: V7,V9:V8
最优剖分: V3,V6:V4	次优剖分: V7,V10:V9
最优剖分: V6,V8:V7	次优剖分: V7,V11:V10
最优剖分: V6,V9:V8	次优剖分: V6,V11:V7
最优剖分: V3,V9:V6	次优剖分: V5,V11:V6
最优剖分: V10,V12:V11	次优剖分: V5,V12:V11
最优剖分: V9,V12:V10	次优剖分: V5,V13:V12
最优剖分: V12,V14:V13	次优剖分: V5,V14:V13
最优剖分: V12,V15:V14	次优剖分: V4,V14:V5
最优剖分: V9,V15:V12	次优剖分: V3,V14:V4
最优剖分: V3,V15:V9	次优剖分: V2,V14:V3
最优剖分: V16,V18:V17	次优剖分: V1,V14:V2
最优剖分: V15,V18:V16	次优剖分: V22,V24:V23
最优剖分: V18,V20:V19	次优剖分: V22,V25:V24
最优剖分: V18,V21:V20	次优剖分: V22,V26:V25
最优剖分: V15,V21:V18	次优剖分: V21,V26:V22
最优剖分: V21,V23:V22	次优剖分: V20,V26:V21
最优剖分: V21,V24:V23	次优剖分: V19,V26:V20
最优剖分: V24,V26:V25	次优剖分: V18,V26:V19
最优剖分: V24,V27:V26	次优剖分: V17,V26:V18
最优剖分: V21,V27:V24	次优剖分: V16,V26:V17
最优剖分: V15,V27:V21	次优剖分: V15,V26:V16
最优剖分: V3,V27:V15	次优剖分: V14,V26:V15
最优剖分: V2,V27:V3	次优剖分: V1,V26:V14
最优剖分: V1,V27:V2	次优剖分: V0,V26:V1
最优剖分: V0,V27:V1	次优剖分: V0,V27:V26

21 个基站最优划分



21 个基站次优划分



21 个基站最优划分——次优划分

```
190167.562885 201606.199615
最优剖分: V1,V3:V2          次优剖分: V0,V2:V1
最优剖分: V1,V4:V3          次优剖分: V0,V3:V2
最优剖分: V1,V5:V4          次优剖分: V3,V5:V4
最优剖分: V6,V8:V7          次优剖分: V0,V5:V3
最优剖分: V5,V8:V6          次优剖分: V5,V7:V6
最优剖分: V1,V8:V5          次优剖分: V5,V8:V7
最优剖分: V0,V8:V1          次优剖分: V8,V10:V9
最优剖分: V0,V9:V8          次优剖分: V5,V10:V8
最优剖分: V9,V11:V10        次优剖分: V0,V10:V5
最优剖分: V9,V12:V11        次优剖分: V11,V13:V12
最优剖分: V9,V13:V12        次优剖分: V11,V14:V13
最优剖分: V13,V15:V14       次优剖分: V14,V16:V15
最优剖分: V13,V16:V15       次优剖分: V11,V16:V14
最优剖分: V13,V17:V16       次优剖分: V10,V16:V11
最优剖分: V9,V17:V13        次优剖分: V10,V17:V16
最优剖分: V0,V17:V9         次优剖分: V10,V18:V17
最优剖分: V0,V18:V17        次优剖分: V10,V19:V18
最优剖分: V0,V19:V18        次优剖分: V0,V19:V10
```

5 0-1 背包问题

5.1 问题描述

给定 n 种物品{1, 2, 3, ...,n}和一背包。物品 i 的重量是 wi，其价值为 vi，背包的容量为 C 问:应如何选择装入背包的物品，使得装入背包中物品在其总重量不超过背包容量 C 的前提下，背包中物品的总价值最大?

5.2 设计思路

可证明背包问题符合最优子结构性质，建立计算 m(i, j)的递归式

- 边界条件：对于只有1个物品——最后1个物品{n}、背包容量为j的子问题<n,j>

$$m(n,j)=\begin{cases} v_n & j \geq w_n \\ 0 & 0 \leq j < w_n \end{cases}$$

容量j大于物品n的重量w_n，n可放入背包，放入后价值为v_n

容量j小于物品n的重量w_n，n无法放入背包，价值为0

- 对子问题<i, j>

$$m(i,j)=\begin{cases} \max\limits_{x_i \in (0,1)} \{m(i+1,j), m(i+1,j-w_i)+v_i\} & j \geq w_i \\ m(i+1,j) & 0 \leq j < w_i \end{cases}$$

i)如果第 i 种物品重量 wi 大于容量 j，第 i 种物品无法放入，可以不考虑，xi=0, 向后考虑其它物品，子问题<i, j>缩减为规模更小的<i+1,j>

ii)如果第 i 种物品重量 wi 小于容量 j，第 i 种物品可以考虑放入，分为 2 种情况

-情况 1:将第 i 种物品放入，对总价值的贡献为 vi，剩余容量为 j-wi ，子问题<i, j>缩减为规模更小的<i+1, j-wi>

-情况 2:第 i 种物品不放入，背包容量仍为 j ，子问题<i, j>缩减为规模更小的<i+1,j >

5.3 实验结果

给出最优方案中，

1. 各个物品是否被放入
2. 物品放入后的背包总重量、总价值

第一组数据：

```
51
340
50 72 16 69 46 9 8 59 49 28 36 1 38 28 2 35 74 71 44 61 58 55 63 59 48 41 39 9
    25 9 32 50 48 19 22 3 19 51 68 9 77 4 72 46 41 20 12 15 52 77 89
14 11 36 17 40 43 14 26 10 48 16 45 41 15 31 21 37 19 48 29 30 50 13 15 9 10
    46 12 49 19 49 8 11 21 11 36 13 8 50 42 45 10 28 16 9 17 18 33 23 18 38
1181
1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0
    0 0 0 1 1 1 0 0 0 1 1 1 Program ended with exit code: 0
```

第二组数据：

```
101
650
61 50 20 14 61 79 12 52 59 20 30 31 33 13 55 12 44 37 67 55 72 74 46 60 16 22
    51 7 18 22 64 35 57 54 63 46 32 18 30 18 4 54 23 13 33 62 23 9 19 21 10 67
    22 73 44 7 22 36 24 49 79 51 12 12 12 18 66 38 42 77 8 25 70 48 56 34 74
    71 31 46 35 46 73 39 13 32 2 74 67 10 37 50 19 41 80 29 7 23 29 50 50
10 39 50 13 33 44 11 46 15 39 12 20 32 11 42 10 30 38 47 49 40 20 18 38 10 35
    28 11 15 28 24 42 44 31 39 36 42 34 19 24 43 26 39 39 36 39 44 26 9 19 26
    18 39 42 15 48 28 11 44 18 8 45 16 47 13 45 50 38 30 45 48 23 50 35 44 35
    18 45 26 20 11 23 26 9 16 43 34 36 24 9 38 47 23 40 30 30 46 38 39 25 40
金额: 1661
背包重量: 639
1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0
    0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
    1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 Program ended with exit
    code: 0
```

6 时间、空间复杂性分析

6.1 最长公共子序列

时间复杂度为 o(mn)，空间复杂度为 o(mn)

6.2 最大字段和

时间复杂度为 o(n)，空间复杂度为 o(n)

6.3 凸多边形最优三角剖分

算法一：

因为进行全遍历时间复杂度为 o(n^3)，空间复杂度为 o(n^2)，时间复杂性较高 o(n^3)，所以当 n 较大时，算法运行时间较长。

算法二：

只选择三个固定位置，作为断点 K 的备选，时间复杂度为 o(n^3)，空间复杂度为 o(n^2)。

6.4 0-1 背包问题

时间复杂度为 o(nc)，空间复杂度为 o(nc)

7 实验总结

在本次实验中，每个任务我们都共同付出了较多的努力。每个人的贡献度为 50%+50%。

本次实验总体较为顺利，较难实现的点在与凸多边形最优三角剖分，其余三个算法都比较顺利实现了。在最长公共子序列算法实现时，需要注意 c 的数组长度是比字符串长度大 1 的，在涉及到有关下标运算的时候需要辨认清楚怎么代入使用。在凸多边形最优三角剖分中，关于如何避免对同一条边的重复运算需要采用科学的方法，我们最后采用的方案是算完整体再减去重复的部分。

改进思路：

1、在 ppt 中提到数组元素 c[i][i]的值仅由 c[i-1][j-1]，c[i-1][i]和 c[i][j-1]这 3 个数组元素的值所确定。对于给定的数组元素 c[i][i]，可以不借助于数组 b 而仅借助于 c 本身在 o(1)时间内确定 c[i]li]的值是由 c[i-1][j-1]，c[i-1][i]和 c[i][j-1]中哪一个值所确定的。在实验中曾根据这一点进行判断，却发现这样的改进是有错误的，因为可能同时出现多个满足的条件，c[i][j]可能为 c[i-1][j-1]+1，也可能与 c[i-1][j]相等，因此这样的判断是不够科学的，最后依然还需要用数组 b 来标记是哪种情况。

2、在凸多边形最优三角剖分中，对于算法二，选取固定位置的点，对于贪心性质有很大的局限性，可以采用随机数的方式在可能断点中随机选取三个点，进行比较计算求出最优划分。

3、关于动态规划的情形中，例如最长公共子序列和 0-1 背包问题中，如果只需要计算最长公共子序列的长度，则算法的空间需求可大大减少。在计算 c[i][j]时，只用到数组 c 的第 i 行和第 i-1 行。因此，用 2 行的数组空间就可以计算出最长公共子序列的长度。进一步的分析还可将空间需求减至 O(min(m,n))。