# 动态规划 Assignment_1

学号 BY1706126
姓名 刘云飞

## 投资问题的 dp 手工求解

### 问题描述

有 8 万元的投资可以投给3 个项目，每个项目在不同投资数额下（以万元为单位）的利润如下表。

| 投资额 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 项目 1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| 项目 2 | 0 | 5 | 15 | 40 | 60 | 70 | 73 | 74 | 75 |
| 项目 3 | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |

写出你所设的状态变量、决策变量、状态转移方程与递推关系式，和手工求解的详细步骤及结果。

### 问题求解

设$\mathrm{f}_k(\mathrm{x})$表示将投资额为x万元投给前k个项目所得到的最高利润，$\mathrm{g}_k(y)$表示对第k个项目投资 y 万元所获得的利润，其中$\mathrm{k}=1,2,3$，$\mathrm{x},\mathrm{y}=0,1,2,3,4,5,6,7,8,\ \mathrm{s.t.}\ \mathrm{y}\le\mathrm{x}$。

状态变量 $\mathrm{f}_k(\mathrm{x})$

决策变量 $\mathrm{g}_k(y)$

状态转移方程 $f(x-y)$

递推关系式 $\mathrm{f}_k(x)=\max\limits_{y=0,1,\dots,x}(g_k(y)+f_{k-1}(x-y))$

递推的详细过程如下：

$\mathrm{f}_1(x)|x$：当只投资一个项目：

| $\mathrm{f}_1(x)\|x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| optimal | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |

$\mathrm{f}_2(8)|x$：当投资两个项目，共投资$\mathrm{x}=8$万元：

| y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $\mathrm{g}_2(y)$ | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| $\mathrm{f}_1(x-y)$ | 75 | 74 | 73 | 70 | 60 | 40 | 15 | 5 | 0 |
| $g_2(y)+f_1(x-y)$ | 75 | 79 | 88 | 110 | 140 | 130 | 110 | 103 | 100 |
| $\mathrm{f}_2(8)$ | | | | | 140 | | | | |

$f_2(7)|x$：当投资两个项目，共投资$x=7$万元：

| y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $g_2(y)$ | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 |
| $f_1(x-y)$ | 74 | 73 | 70 | 60 | 40 | 15 | 5 | 0 |
| $g_2(y)+f_1(x-y)$ | 74 | 78 | 85 | 100 | 120 | 105 | 100 | 98 |
| $f_2(8)$ | | | | | 120 | | | |

$f_2(6)|x$：当投资两个项目，共投资$x=6$万元：

| y | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $g_2(y)$ | 0 | 5 | 15 | 40 | 80 | 90 | 95 |
| $f_1(x-y)$ | 73 | 70 | 60 | 40 | 15 | 5 | 0 |
| $g_2(y)+f_1(x-y)$ | 73 | 75 | 75 | 80 | 95 | 95 | 95 |
| $f_2(8)$ | | | | | 95 | | |

$f_2(5)|x$：当投资两个项目，共投资$x=5$万元：

| y | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $g_2(y)$ | 0 | 5 | 15 | 40 | 80 | 90 |
| $f_1(x-y)$ | 70 | 60 | 40 | 15 | 5 | 0 |
| $g_2(y)+f_1(x-y)$ | 70 | 65 | 55 | 50 | 85 | 90 |
| $f_2(8)$ | | | | | | 90 |

$f_2(4)|x$：当投资两个项目，共投资$x=4$万元：

| y | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $g_2(y)$ | 0 | 5 | 15 | 40 | 80 |
| $f_1(x-y)$ | 60 | 40 | 15 | 5 | 0 |
| $g_2(y)+f_1(x-y)$ | 60 | 45 | 30 | 45 | 80 |
| $f_2(8)$ | | | | | 80 |

$f_2(3,2,1,0)|x$：当投资两个项目，共投资$x=3,2,1,0$万元：

| y | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $g_2(y)$ | 0 | 5 | 15 | 40 | 0 | 5 | 15 | 0 | 5 | 0 |
| $f_1(x-y)$ | 40 | 15 | 5 | 0 | 15 | 5 | 0 | 5 | 0 | 0 |
| $g_2(y)+f_1(x-y)$ | 40 | 20 | 20 | 40 | 15 | 10 | 15 | 5 | 5 | 0 |
| $f_2(8)$ | 40 | | | | | | 15 | 5 | | 0 |

$f_3(8)|x$：当投资三个项目，共投资$x=8$万元：

| y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $g_3(y)$ | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |
| $f_2(x-y)$ | 140 | 120 | 95 | 90 | 80 | 40 | 15 | 5 | 0 |
| $g_3(y)+f_2(x-y)$ | 140 | 124 | 121 | 130 | 125 | 90 | 66 | 57 | 53 |
| $f_2(8)$ | 140 | | | | | | | | |

由此得到最优解为 140， 向第一个项目、第二个项目、第三个投资 4，4，0 万元。

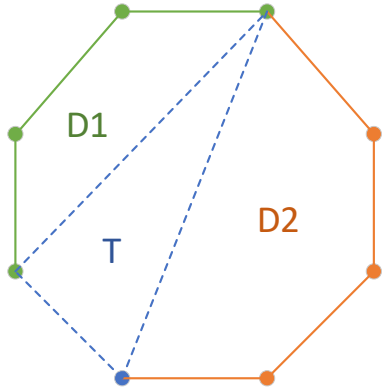
# 凸 8 边形的三角割分

## 问题描述

一个凸 8 边形P 的顶点顺时针为{$v_1$，$v_2$，…，$v_8$}，任意两顶点间的线段的权重由矩阵D 给出。若$v_i$与 $v_j$与是P上不相邻的两个顶点，则线段$v_iv_j$称为P的一条弦。求P的一个弦的集合T，使得T中所有的弦恰好将P分割成互不重叠的三角形，且各三角形的权重之和为最小（一个三角形的权重是其各边的权重之和）。
要求：写出递推关系式、伪代码和程序相关说明，并分析时间复杂性。

## 问题分析

将凸多边形的三角割分问题可以看作每完成一次三角形割分，剩下的部分，可以看作两个多边形的三角割分的子问题（所分割的三角形不是相邻的三个顶点，否则可以看作为剩下的一个凸多边形的三角割的分子问题），如下图所示：




所以如果取得的 T 区域的三角形为凸 8 边形的一个最优的三角割分，那么当且仅当剩下的这两个割分的区域 D1 与 D2 求得的凸多边形割分的结果也是最优的。所以当选取其中两个顶点$v_i$和$v_j$来求解凸多边形的最优三角割时，有如下推导式：

$$T_{i,j} = optimal(T_{i,k} + T_{k+1,j} + weight(v_{i-1}, v_j, v_k))$$

其中$T_{i-1,k}$表示 D1 的最优解，$T_{k,j+1}$表示 D2 的最优解，本案例中是求解三角形的权重之和最小，考虑到迭代的边界条件可以得到下面的递推关系式：

$$T_{i,j} = \begin{cases} 0 & if\ i == j \\ T_{i,k} + T_{k+1,j} + min(v_{i-1}, v_j, v_k) & otherwise \end{cases}$$

# 伪代码

伪代码的展示如下：

定义一个 N×N 的二维方阵 T，此处 N=8

1. 初始化矩阵 T，使全部元素都为 0
2. 初始化与矩阵 T 同样形状的矩阵 S 用于记录最终相连的边
3. 子问题的规模为 N-1，进行如下循环：

```
For r = [1: 1: N] do            // r 表示当前子问题的长度，即点的个数
    For i = [1: 1: N-r] do      // N-r 表示最后一个子问题的前边界，三角形的第一个点
        j ← i + r               // 前边界为 r，链长度为 r 的后边界，三角形的第二个点
        T[i, j] ← T[i+1, j] + weight(i-1, i, j)     // 初始化 T[i, j]，此时 k==i
        S[i, j] ← i                                 // 初始化 S[i, j]，暂时记录为 i
        For k = [i + 1: 1 : j] do                   // 在前边界 i 到后边界 j 之间搜索更新最优解
            tmp ← T[i+1, j] + T[k+1, j] + weight(i-1, k, j)
            if tmp < T[i, j] then                   // 更新最小值和顶点
                T[i, j] ← tmp
                S[i, j] ← k
            EndIf
        EndFor
    EndFor
EndFor
```

4. 最后得到的 T[1, N-1]就是累加得到的凸多边形的三角割分的最小权重

注：其中 weight(i, j, k)是求解以$v_i$，$v_j$，$v_k$为定点的三角形三条边的权重和

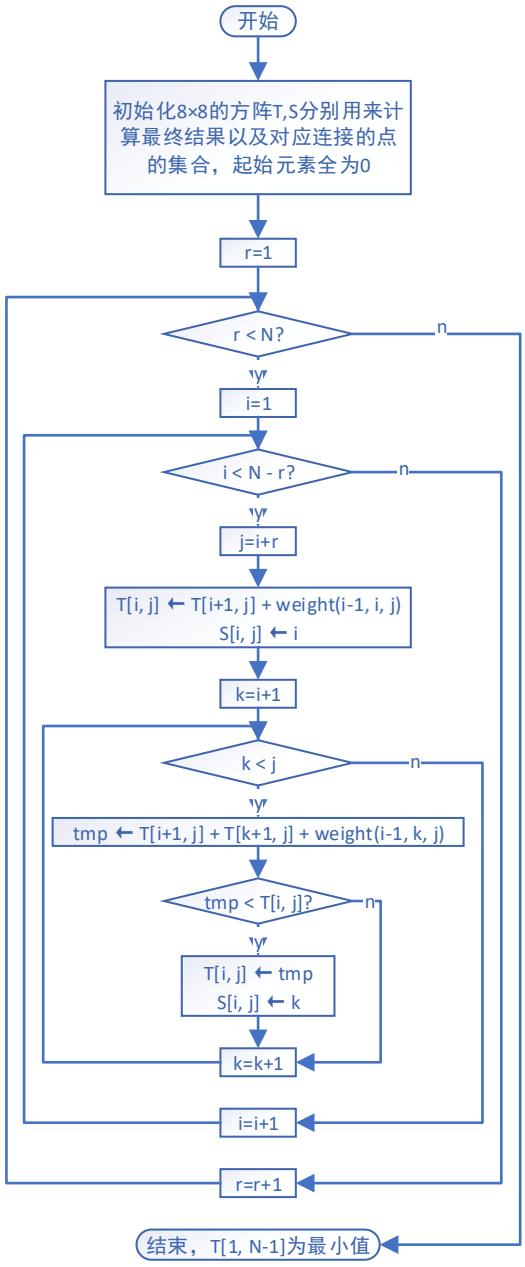
# 算法分析

通过上述的伪代码可以分析算法的循环层数及其循环次数如下

r 层执行 N-1 次→i 层执行 N-r-1 次，其中有三个赋值和一个循环→k 层循环 r-1，其中最大包含三个赋值。所以算法复杂公式可以如下：

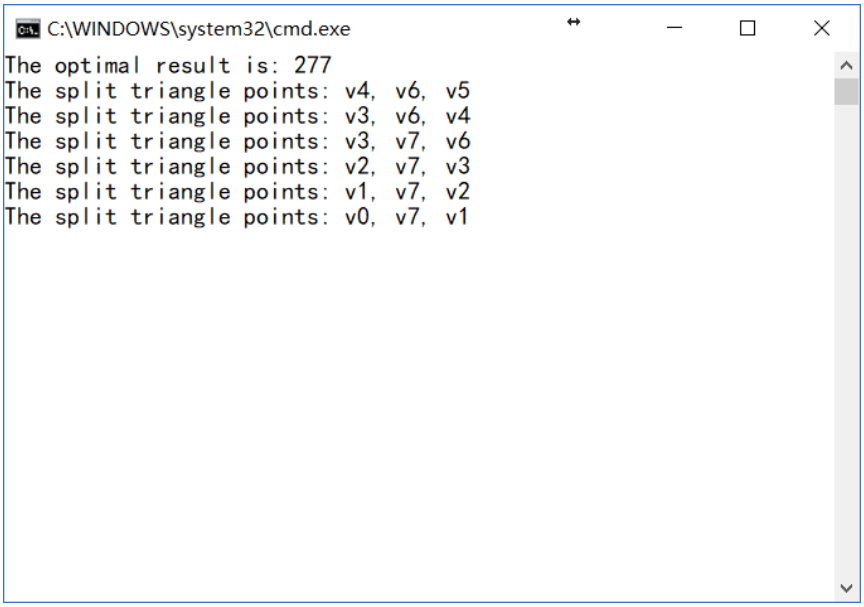
$$\omega = \sum_{r=1}^{N-1} \sum_{i=1}^{N-r-1} (3 + \sum_{k=1}^{r-1} 3) = O(n^3)$$

# 流程图


开始
初始化8×8的方阵T,S分别用来计算最终结果以及对应连接的点的集合，起始元素全为0
r=1
r < N?
n
y
i=1
i < N - r?
n
y
j=i+r
T[i, j] ← T[i+1, j] + weight(i-1, i, j)
S[i, j] ← i
k=i+1
k < j
n
y
tmp ← T[i+1, j] + T[k+1, j] + weight(i-1, k, j)
tmp < T[i, j]?
n
y
T[i, j] ← tmp
S[i, j] ← k
k=k+1
i=i+1
r=r+1
结束，T[1, N-1]为最小值

# 处理结果

通过运行附件 1 的算法对应的可执行程序可以得到对应结果如下，或者点击这里来查看更多详细信息：




对应的定点序号从 0~7，共八个顶点。分割的结果如下图：