

# 选课策略问题

## 问题重述

课号	课名	学分	所属类别	先修课要求
1	微积分	5	数学	
2	线性代数	4	数学	
3	最优化方法	4	数学、运筹学	微积分、线代
4	数据结构	3	数学、计算机	计算机编程
5	应用统计	4	数学、运筹学	微积分、线代
6	计算机模拟	3	计算机、运筹学	计算机编程
7	计算机编程	2	计算机	
8	预测理论	2	运筹学	应用统计
9	数学实验	3	计算机、运筹学	微积分、线代

必须完成的要求：至少选择两门数学课、三门运筹学课和两门计算机课。

## 变量定义：

$x_i$ : 0 或者 1。0 表示不选择该课程, 1 表示选择该课程。 $i = 1, 2, \dots, 9$

$c_i$ : 表示第  $i$  门课程的学分。 $i = 1, 2, \dots, 9$

## 问题一：

选修课门数最少，应学习哪些课程？

$$\min \quad number = \sum_{i=1}^{i=9} x_i$$

$$\text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \geq 2, \\ x_4 + x_6 + x_7 + x_9 \geq 2, \\ x_3 + x_5 + x_6 + x_8 + x_9 \geq 3, \\ x_1 + x_2 \geq 2 \cdot x_3, \\ x_7 \geq x_4, \\ x_1 + x_2 \geq 2 \cdot x_5, \\ x_7 \geq x_6, \\ x_5 \geq x_8, \\ x_1 + x_2 \geq 2 \cdot x_9, \\ x_1, x_2, \dots, x_9 = 0 \text{ or } 1 \end{cases} \Rightarrow \begin{cases} -x_1 - x_2 - x_3 - x_4 - x_5 \leq -2, \\ -x_4 - x_6 - x_7 - x_9 \leq -2, \\ -x_3 - x_5 - x_6 - x_8 - x_9 \leq -3, \\ -x_1 - x_2 + 2 \cdot x_3 \leq 0, \\ -x_7 + x_4 \leq 0, \\ -x_1 - x_2 + 2 \cdot x_5 \leq 0, \\ -x_7 + x_6 \leq 0, \\ -x_5 + x_8 \leq 0, \\ -x_1 - x_2 + 2 \cdot x_9 \leq 0, \\ x_1, x_2, \dots, x_9 = 0 \text{ or } 1 \end{cases}$$

code:

```
clc, clear
f = ones(9, 1);
intcon = 1:9;
a = [-1, -1, -1, -1, -1, 0, 0, 0, 0;
     0, 0, 0, -1, 0, -1, -1, 0, -1;
     0, 0, -1, 0, -1, -1, 0, -1, -1;
     -1, -1, 2, 0, 0, 0, 0, 0, 0;
     0, 0, 0, 1, 0, 0, -1, 0, 0;
     -1, -1, 0, 0, 2, 0, 0, 0, 0;
     0, 0, 0, 0, 0, 1, -1, 0, 0;
     0, 0, 0, 0, -1, 0, 0, 1, 0;
     -1, -1, 0, 0, 0, 0, 0, 0, 2];
b = [-2; -2; -3; 0; 0; 0; 0; 0; 0];
lb = zeros(9, 1);
ub = ones(9, 1);
x = intlinprog(f, intcon, a, b, [], [], lb, ub);
```

x

求解：

计算机给出了一种解法，实际上肯定不是一种解法

$x_1 = x_2 = x_3 = x_6 = x_7 = x_9 = 1$ ，其余为 0.

课号	课名	学分	所属类别	先修课要求
1	微积分	5	数学	
2	线性代数	4	数学	
3	最优化方法	4	数学、运筹学	微积分、线代
6	计算机模拟	3	计算机、运筹学	计算机编程
7	计算机编程	2	计算机	
9	数学实验	3	计算机、运筹学	微积分、线代

## 问题二：

选修课程最少，且学分尽量多，应学习哪些课程？

$$\begin{aligned} \min \quad & number = \sum_{i=1}^{i=9} x_i \\ \max \quad & scores = \sum_{i=1}^9 x_i \cdot c_i \\ & \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \geq 2, \\ x_4 + x_6 + x_7 + x_9 \geq 2, \\ x_3 + x_5 + x_6 + x_8 + x_9 \geq 3, \\ x_1 + x_2 \geq 2 \cdot x_3, \\ x_7 \geq x_4, \\ x_1 + x_2 \geq x_5, \\ x_7 \geq x_6, \\ x_5 \geq x_8, \\ x_1 + x_2 \geq 2 \cdot x_9 \end{cases} \end{aligned}$$

我们发现第一问只给出了一种答案，而实际上应该在第一问的多种方案中，寻找学分最多的解。

单独规划出来一个目标，因为学分的比重比较重：

$$\begin{aligned} \max \quad & comprehensive \ scores = -0.5 \cdot number + 0.5 \cdot scores \\ \min \quad & -comprehensive \ scores = 0.5 \cdot number - 0.5 \cdot scores \end{aligned}$$

代入化简

规划的目标范围不变：发现效果不好！！

改变策略，由于数据范围为 9，枚举解答，有 9! 种可能。

code:

```
#include <bits/stdc++.h>
using namespace std;
const int maxn = 15;
int x[maxn];
int ans[maxn];
int c[maxn] = {0, 5, 4, 4, 3, 4, 3, 2, 2, 3};
bool judge(int* number){
    if(x[1]+x[2]+x[3]+x[4]+x[5]<2) return false;
    if(x[4]+x[6]+x[7]+x[9]<2) return false;
    if(x[3]+x[5]+x[6]+x[8]+x[9]<3) return false;
    if(x[1]+x[2]-2*x[3]<0) return false;
    if(x[7]-x[4]<0) return false;
    if(x[1]+x[2]-2*x[5]<0) return false;
    if(x[7]-x[6]<0) return false;
    if(x[5]-x[8]<0) return false;
```

```
        if(x[1]+x[2]-2*x[9]<0) return false;
        return true;
    }
    int cal(int* num){
        int ans = 0;
        for(int i=1; i<10; i++){
            ans += c[i]*num[i];
        }
        return ans;
    }
    int main()
    {
        memset(ans, -1, sizeof(ans));
        int score = -1;
        int class_number = 6;
        for(int s=0; s<(1<<10); s++){
            class_number = 0;
            for(int i=0; i<9; i++){
                if(s&(1<<i)){
                    x[i+1] = 1;
                    class_number++;
                }
                else x[i+1] = 0;
            }
            if(judge(x)&&class_number == 6){
                int temp = cal(x);
                //printf("%d\n", temp);
                if(temp>score){
                    score = temp;
                    for(int i=1; i<10; i++){
                        ans[i] = x[i];
                    }
                }
            }
        }
        for(int i=1; i<10; i++){
            printf("x%d: %d ", i, ans[i]);
        }
        printf("\n");
        printf("%d\n", score);

        return 0;
    }
}
```

很容易的就得到了答案：

课号	课名	学分	所属类别	先修课要求
1	微积分	5	数学	
2	线性代数	4	数学	
3	最优化方法	4	数学、运筹学	微积分、线代
6	计算机模拟	3	计算机、运筹学	计算机编程
7	计算机编程	2	计算机	
5	应用统计	4	数学、运筹学	微积分、线代

最大的学分为 22.