

重 庆 大 学

学 生 实 验 报 告

实验课程名称 数学实验

开课实验室 DS1401

组员 1 姓 名 马梓恒 学 号 20233124

组员 2 姓 名 周宏仰 学 号 20232647

组员 3 姓 名 李宇聪 学 号 20232137

开 课 时 间 2024 至 2025 学年第 一 学期

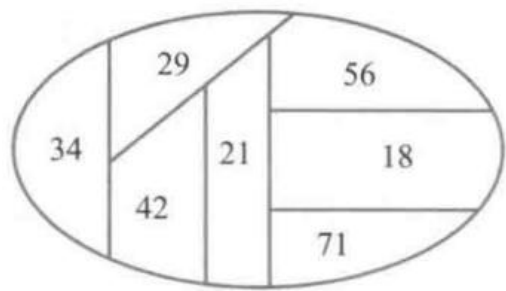
总 成 绩	
-------	--

数 统 学 院 制

课程名称	数学实验	实验项目名称	数学规划	实验项目类型				
				验证	演示	综合	设计	其他
指导教师	肖剑	成绩				√		

题目 1

一家出版社准备在某市建立两个销售代理点，向 7 个区的大学生售书，每个区的大学生数量(单位:千人)已经表示在图上。每个销售代理点只能向本区和一个相邻区的大学生售书，这两个代理点应该建在何处，才能使所能供应的大学生的数量最大?建立该问题的数学规划模型并求解。



程序

```
model:
max=63*x12+76*x13+71*x23+50*x24+85*x25+63*x34+77*x45+39*x46+92*x47+74*x56+89*x67;
x12+x13+x23+x24+x25+x34+x45+x46+x47+x56+x67<=2;
x12+x13<=1;
x12+x23+x24+x25<=1;
x13+x23+x34<=1;
x25+x45+x56<=1;
x46+x56+x67<=1;
x47+x67<=1;
@bin(x12);@bin(x13);
@bin(x23);@bin(x24);@bin(x25);
@bin(x34);
@bin(x45);@bin(x46);@bin(x47);
@bin(x56);
@bin(x67);
```

结果

找到最优解。

Global optimal solution found.

Objective value:	177.0000
Objective bound:	177.0000

Infeasibilities:	0.000000
Extended solver steps:	0
Total solver iterations:	0
Elapsed runtime seconds:	0.04

Model Class:	PILP
--------------	------

Total variables:	11
Nonlinear variables:	0
Integer variables:	11
Total constraints:	8
Nonlinear constraints:	0
Total nonzeros:	39
Nonlinear nonzeros:	0

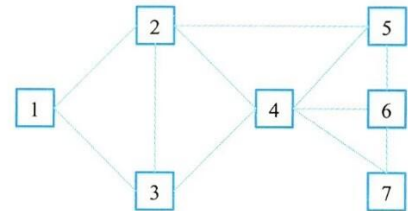
Variable	Value	Reduced Cost
X12	0.000000	-63.00000
X13	0.000000	-76.00000
X23	0.000000	-71.00000
X24	0.000000	-50.00000
X25	1.000000	-85.00000
X34	0.000000	-63.00000
X45	0.000000	-77.00000
X46	0.000000	-39.00000
X47	1.000000	-92.00000
X56	0.000000	-74.00000
X67	0.000000	-89.00000
Row	Slack or Surplus	Dual Price
1	177.0000	1.000000
2	0.000000	0.000000

3	1.000000	0.000000
4	0.000000	0.000000
5	1.000000	0.000000
6	0.000000	0.000000
7	1.000000	0.000000
8	0.000000	0.000000

分析

将大学生数量为 34, 29, 42, 21, 56, 18, 71 的区分别标号为 1, 2, 3, 4, 5, 6, 7 区，画出区与区之间的相邻关系图（图 1）：

记 r_i 为第 i 区的大学生人数，用 0-1 变量， $x_{ij}=1$ 来表示（ i, j ）区的大学生由一个销售代理点供应图书（ $i < j$ 且 i, j 相邻），否则 $x_{ij}=0$ 。建立该问题的整数线性规划模型。



$$\max \sum_{i,j \text{ 相邻}} (r_i + r_j) x_{ij}$$

$$\begin{aligned} \text{s. t. } & \sum_{i,j} x_{ij} \leq 2 \\ & \sum_j x_{ij} + \sum_j x_{ji} \leq 1, \forall i \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

用 lingo 求解最大值，即为最大学生数量；此时 $x_{ij}=1$ 中的同一个 x 的 i, j 所对应的销售代理点之一即为所要选区（ i 区和 j 区均可，但只取一个）。

题目 2

某部门在今后五年内考虑给下列项目投资。已知：

项目 A：从第一年到第四年每年年初需要投资，并于次年末回收本利 115%；

项目 B：从第三年初需要投资，到第五年末能回收本利 125%，但规定最大投资额不超过 4 万元；

项目 C：第二年初需要投资，到第五年末能回收本利 140%，但规定最大投资额不超过 3 万元；

项目 D：五年内每年年初可购买公债，于当年末归还，并加利息 6%。

该部门现有资金 10 万元，问应如何确定给这些项目每年的投资额，使到第五年末拥有的资金的本利总额为最大

程序

```
model:
max=1.15*x41+1.40*x23+1.25*x32+1.06*x54;
x11+x14=10;
-1.06*x14+x21+x23+x24=0;
-1.15*x11-1.06*x24+x31+x32+x34=0;
-1.15*x21-1.06*x34+x41+x44=0;
```

-1.15*x31-1.06*x44+x54=0;
x32<=4;
x23<=3;

结果

Global optimal solution found.
Objective value: 14.37500
Infeasibilities: 0.000000
Total solver iterations: 1
Elapsed runtime seconds: 0.03

Model Class: LP

Total variables: 11
Nonlinear variables: 0
Integer variables: 0

Total constraints: 8
Nonlinear constraints: 0

Total nonzeros: 24
Nonlinear nonzeros: 0

Variable	Value	Reduced Cost
X41	4.500000	0.000000
X23	3.000000	0.000000
X32	4.000000	0.000000
X54	0.000000	0.000000
X11	7.169811	0.000000
X14	2.830189	0.000000
X21	0.000000	0.000000
X24	0.000000	0.3036000E-01
X31	0.000000	0.000000
X34	4.245283	0.000000
X44	0.000000	0.2640000E-01

Row	Slack or Surplus	Dual Price
1	14.37500	1.000000
2	0.000000	1.401850
3	0.000000	1.322500
4	0.000000	1.219000
5	0.000000	1.150000

6	0.000000	1.060000
7	0.000000	0.310000E-01
8	0.000000	0.775000E-01

分析

① 确定变量设 x_{ij} , ($i=1, 2, 3, 4, 5$; $j=1, 2, 3, 4$) 表示第 i 年年初给项目 j 的投资额, 将变量列于表中:

项目 \ 年份	1	2	3	4	5
1 (A)	x_{11}	x_{21}	x_{31}	x_{41}	
2 (B)			x_{32}		
3 (C)		x_{23}			
4 (D)	x_{14}	x_{24}	x_{34}	x_{44}	x_{54}

② 确定每年年初的投资额. 由于投资额应不大于手中拥有的资金额, 而项目 D 每年都可以投资, 且当年末即能回收本息, 所以该项目每年应把资金全部投出去, 手中不应当有剩余的呆滞资金, 因此每年的投资额和本利和为:

第一年, 该部门年初拥有 10 万元, 所以 $x_{11}+x_{14}=10$

第二年, 因第一年给项目 A 的投资要到该年末才能回收, 所以该部门在该年初拥有的资金额仅为项目 D 在第一年回收的本利和 $x_{14}(1+6\%)$, 于是第二年的投资分配是 $x_{21}+x_{23}+x_{24}=x_{14}(1+6\%)$

第三年, 第三年初的资金额是从项目 A 的第一年投资及项目 D 的第二年投资中回收的本利总和 $x_{11}(1+15\%)$ 及 $x_{24}(1+6\%)$, 于是该年的资金分配为 $x_{31}+x_{32}+x_{34}=x_{11}(1+15\%)+x_{24}(1+6\%)$

第四年, 同以上分析, 可得年初的资金分配情况为 $x_{41}+x_{44}=1.15x_{21}+1.06x_{34}$

第五年, 年初的资金分配为 $x_{51}=1.15x_{31}+1.06x_{44}$

此外, 由于对项目 B、C 的投资有限额的规定, 有: $x_{32} \leq 4$, $x_{23} \leq 3$

③ 第五年末该部门拥有的资金总额为 $1.40x_{23}+1.25x_{32}+1.15x_{41}+1.06x_{34}$, 用 lingo 求解其最大值, 同时会返回 x_{ij} ($i=1, 2, 3, 4, 5$; $j=1, 2, 3, 4$) 的值, 则投资方案解出。

题目三

请自行查询某商业银行的整存整取年利率, 填入下表:

一年期	二年期	三年期	五年期
1.35	1.45	1.75	1.80

现有 1 笔本金, 准备 33 年后使用, 若此期间利率不变, 问应该采用怎样的存款方案?

程序

```
def calculate_best_option(principal, years, rates):
    # 初始化动态规划数组, 存储每年结束时的最大收益和存款策略
    dp = [0] * (years + 1)
    strategy = [None] * (years + 1)
```

```

# 遍历每一年的存款选择
for year in range(1, years + 1):
    for term, rate in rates.items():
        if year >= term:
            # 计算当前的收益
            future_value = principal * (1 + rate / 100)

            if future_value > dp[year]:
                dp[year] = future_value

            # 记录存款策略
            strategy[year] = (term, future_value)

    return dp, strategy

def get_saving_plan(principal, years, rates):
    # 计算最佳选项
    dp, strategy = calculate_best_option(principal, years, rates)

    # 输出最佳收益和存款步骤
    total_profit = dp[years]
    plan = []
    current_year = years

    while current_year > 0:
        term, future_value = strategy[current_year]
        plan.append((current_year - term + 1, term))
        current_year -= term

    plan.reverse() # 反转计划顺序，以便从开始到结束展示

    return total_profit, plan

# 利率设置
rates = {
    1: 1.35,
    2: 1.45,
    3: 1.75,
    5: 1.80
}

# 本金和投资年份
principal = 1 # 初始本金设为1，用于计算总收益
years = 33

# 计算最佳存款方案

```

```
total_profit, saving_plan = get_saving_plan(principal, years, rates)

# 输出结果
print(f"总收益（本金 1 元，33 年后）：{total_profit:.2f}元")
print("存款策略（起始年，存款年限）：")
for start_year, term in saving_plan:
    print(f"从第 {start_year} 年存款 {term} 年")
```

结果

总收益（本金 1 元，33 年后）：1.02 元

存款策略（起始年，存款年限）：

从第 1 年存款 3 年

从第 4 年存款 5 年

从第 9 年存款 5 年

从第 14 年存款 5 年

从第 19 年存款 5 年

从第 24 年存款 5 年

从第 29 年存款 5 年

分析

这段代码实现了一个动态规划算法，用于计算基于不同存款期限和利率的最佳存款策略，以最大化投资收益。函数 `calculate_best_option` 初始化了两个数组：`dp` 用于存储每年结束时的最大收益，`strategy` 用于记录对应的存款策略。该函数遍历每个年份及其可选的存款期限和利率，计算并更新当前年份的最大收益。

另一个函数 `get_saving_plan` 利用 `calculate_best_option` 输出最终的总收益和具体的存款方案。代码中定义了一些固定的利率选项，并设定本金为 1 元，投资年限为 33 年。最后通过打印输出，展示了在指定条件下的总收益和详细的存款步骤。这种方法有效地将时间和收益结合，为用户提供了一种系统化的投资策略选择。

选做题

A 公司面临破产，只余下 100 种物品，表 1 中给出了每种物品的数量，现有 1000 名公司债权人，表格中给出了债权人对不同物品的偏好（数值越大越喜欢），要求你们对这些资产进行处置，应该如何安排呢？（如果讨论有困难的同学，可以把问题的规模适当缩小，如只选取前 10 种物品，前 10 名债权人来讨论）

程序

```
% 读取 Excel 文件中的数据
data = readtable('1.xls');
```



```

% 提取物品编号和数量
items = table2array(data(1, 3:end)); % 提取物品编号 (C 列及后续)
quantities = table2array(data(2, 3:end)); % 提取物品数量 (C 列及后续)

% 提取债权人的偏好表和名字
preferences = table2array(data(3:end, 3:end)); % 提取偏好值表
creditor_names = table2array(data(3:end, 2)); % 提取债权人名字
[num_creditors, num_items] = size(preferences); % 获取债权人数量和物品数量

% 初始化分配矩阵，行对应债权人，列对应物品
allocation = zeros(num_creditors, num_items);

% 为每个物品进行分配
for idx = 1:num_items
    total_preference = sum(preferences(:, idx)); % 计算该物品的总偏好值
    if total_preference > 0 % 如果有偏好值
        % 计算每个债权人分配的数量
        for creditor = 1:num_creditors
            allocation(creditor, idx) = floor(quantities(idx) * (preferences(creditor, idx) /
total_preference));
        end

        % 确保分配总数不超过该物品的数量
        while sum(allocation(:, idx)) > quantities(idx)
            % 按偏好降序调整
            [~, sorted_indices] = sort(preferences(:, idx), 'descend');
            for k = sorted_indices'
                if allocation(k, idx) > 0
                    allocation(k, idx) = allocation(k, idx) - 1;
                    if sum(allocation(:, idx)) <= quantities(idx)
                        break;
                    end
                end
            end
        end
    end
end

% 定义一个宽 100，高 1000 的矩阵，所有数据都为 0
matrix = zeros(1000, 100);

% 输出物品分配结果到命令行
disp(' 物品分配结果: ');
for creditor = 1:num_creditors

```

```

    creditor_name = creditor_names(creditor); % 债权人名字

% 找到分配的物品编号和数量
allocated_items = find(allocation(creditor, :));
item_ids = items(allocated_items);
item_counts = allocation(creditor, allocated_items); % 每种物品的分配数量

% 输出结果
for i = 1:length(item_ids)
    disp([creditor_name, num2str(item_ids(i)), num2str(item_counts(i))]);
    matrix(creditor, item_ids(i)) = item_counts(i);
end
end

% 将矩阵写入 Excel 文件
writematrix(matrix, 'outputData.xlsx');

```

结果

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
剩余物品数量排序									
1	2	3	4	5	6	7	8	9	10
23	17	12	10	8	8	6	5	4	4

分析

这段 MATLAB 代码主要用于从 Excel 文件中读取物品及其分配给债权人的数据。首先，它提取了物品编号、数量以及债权人的偏好值和名字。然后，通过计算每个物品的总偏好值，依据偏好比例分配物品给债权人，并确保分配的总数量不超过实际数量。最后，代码输出分配结果，并将结果写入一个新的 Excel 文件。

备注：

- 1、一门课程有多个实验项目的，应每一个实验项目一份，课程结束时将该课程所有实验项目内页与封面合并成一个电子文档上交。