
学校	湖南大学信息科学与工程学院
队员姓名	1. 甘毅辉
	2. 李峥嵘
	3. 李俊杰

摘 要：

针对债券组合投资问题，利用 `scipy` 来解决的非线性规划问题。此方法是解决非线性规划的很经典的方法。对于所得的结果，此算法有较好的可信度。

问题一中，根据题意我们可以得到一系列必要的约束；分析了连续两年之间可用于投资的本金的关系，并给出计算公式；最后通过库函数递归，得到最大化收益；

问题二中，在问题一的基础上，获得银行贷款；我们分析了各项投资项目的年利率，绝大部分项目的年利率都要大于银行利息率（只有一个例外）；为了简化算法复杂性，决定从银行直接获取最大贷款；并且修改了问题一中的约束条件；最后通过同样的算法，得到最大收益；

问题三中，依照题干，修改问题一中的约束条件。问题三中要求我们的收益大于问题二的收益，所以我们首先固定贷款金额来建立模型并计算出收益；然后通过二分法，手动调整贷款金额直至收益大于等于第二问中的收益。此问题收益的求解与问题一相同。

关键字：非线性规划，`scipy`

目录

1. 问题重述.....	3
2. 模型假设.....	3
3. 符号说明.....	3
4. 问题的分析.....	5
4.1 问题一.....	5
4.1.1 问题描述和分析.....	5
4.1.2 模型建立.....	5
4.1.3 模型求解.....	6
4.2 问题二.....	9
4.2.1 问题描述和分析.....	9
4.2.2 模型建立.....	10
4.2.3 模型求解.....	10
4.3 问题三.....	13
4.3.1 问题描述和分析.....	13
4.3.2 模型建立.....	13
4.3.3 模型求解.....	15
附录 A 程序代码.....	18
1.问题一.....	18
2.问题二.....	21
3.问题三.....	24

1. 问题重述

2020 年，新冠肺炎全球蔓延，严重影响了世界福祉。为刺激经济发展，M 国计划发行各类代号为“拯救地球”的国家紧急债券（SEB）。债券到期年限有 2、3、4、5、9、12、15、18、20、25 年等十种，债券每年初发行，发行时间为 2021–2048 年。债券每年末兑现，兑现有效时间为 2023–2050 年。债券服务范围涵盖医药健康、交通运输、科技研发、装备制造、国民福利五大领域。债券到期后，允许债券购买者所在的地方政府收取税率不等的税收。债券类型、到期年限、收益率、税收比率等见表 1。

未来教育基金会（FFE）计划从 2021 年开始，未来 30 年内每年初向 M 国部分高等院校捐助一笔金额固定的经费以帮助贫困学生。未来教育基金会现有 8000 万美元资金可以从事债券投资。为谨慎起见，FFE 基金会委托风险机构对各类债券进行了风险等级评估。同时，FFE 基金会还制定了投资要求：

1. 每个行业的债券购买总金额不得少于债券购买总金额的 10%；
2. 所购债券的平均风险等级不得低于 2.5，等级数字越大，风险越低；
3. 可重复投资。当债券到期后，本金和收益只要投资在 2050 年及之前本息能够回收就可以进行再次投资；
4. 要求所购买的债券的平均到期年限不超过 8 年。

问题 1：由于投资债券时间跨越时间长，未来教育基金会需要有一个 2021–2048 的投资债券详细方案，现委托你作为他们的经济顾问。帮助未来教育基金会制定这个投资方案，以更大程度的惠及贫困学生。

问题 2：经过努力，未来教育基金会在一家银行借到了一笔年利率为 2% 的一次性长期优惠贷款，2020 年年末放贷，不超过 5000 万美元。银行要求从 2021 年起每年年终必须支付利息。请结合这种情况，调整你的投资方案。

问题 3：未来教育基金会为了帮助疫情防控，决定投资向医药健康和科技研发上倾斜。原则上规定：医药健康的投资额，尽量与科技研发相等，尽量是交通运输的 3 倍，尽量是装备制造的 2 倍，尽量是国民福利的 4 倍。捐助给高校的金额尽量不低于第 2 问的金额。为了达到这些要求，未来教育基金会决定向银行追加一次性银行贷款，请问最少需要追加多少？

2. 模型假设

- 假设 1：不考虑通货膨胀；
假设 2：债券风险不随时间改变；
假设 3：投资时间内不发生世界性经济危机；

3. 符号说明

符号	符号说明
i	年份
j	题干表 1 中的项目
k	题干表 1 中的行业

X	每年对每个项目的投资
Yield	每个项目的收益率
Profit	每年对每个项目的收益
F	各行业风险
Y	各项目到期年份
base_money	每年可用于投资的本金
year_invest	每年用于投资的资金
industry_invest	行业总投资
student	帮助贫困生的资金
SUM	矩阵/向量元素和函数
Interest	偿还的贷款利息
Loan	银行贷款
s	投资倍率

取值:

$$\left\{ \begin{array}{l}
 i = 1, 2, \dots, 30 \\
 j = 1, 2, \dots, 15 \\
 k = 1, 2, \dots, 5 \\
 Yield = \{yield_j \mid \dots\} \\
 X = \{x_{i,j}\} \\
 Profit = \{profit_{i,j}\} \\
 F = \{f_k \mid 5, 2, 4, 3, 1\} \\
 Y = \{y_j \mid 2, 9, \dots, 5, 18\} \\
 Base_money = \{8000, \dots, base_money_i, base_money_{i+1}, \dots\} \\
 Year_invest = \{year_invest_i\} = sum(\{x_{i,j} \mid j \leq 15\}) \\
 Industry_invest = \{industry_invest_k\} = sum(\{x_{i',j} \mid i' \leq 30, j \in k\}) \\
 student = C \\
 Interest = C \\
 Loan = C
 \end{array} \right.$$

表 1 各种债券类型

行业编号 K	债券领域	风险	税率	到期年份	到期收益率 yield	投资编号 J
1	医药健康	5	0	2	4.45	1
		5	0	9	39.89	2
		5	0	20	232.11	3
2	交通运输	2	20	3	8	4
		2	20	12	58.27	5
		2	20	25	300.54	6
3	科技研发	4	0	4	12.55	7

		4	0	15	180.09	8
		4	0	20	232.11	9
		3	10	4	12.55	10
4	装备制造	3	10	9	39.89	11
		3	10	18	206.12	12
		1	30	2	4.45	13
5	国民福利	1	30	5	18.77	14
		1	30	18	206.12	15

4. 问题的分析

4.1 问题一

4.1.1 问题描述和分析

根据题目中的描述，基金会现有 8000 万美元可以进行投资，并且该基金会未来 30 年内每年初向 M 国部分高等院校捐助一笔金额固定的经费以帮助贫困学生。基金会的目的是在不亏损的情况下，最大程度帮助贫困生。该问题就变成了——制定一个 30 年的投资计划，使得总收益最大化——非线性规划问题；因为收益越多，可以用来帮助贫困生的资金就越多。

4.1.2 模型建立

问题 1 中，要求我们得到满足一系列约束条件时收益最大的投资组合。若项目 j 在 i 年产生了收益，我们便会记录在 $profit_{i,j}$ 中，总收益便是每次收益之和（本文提到的收益均指的是税后收益）。这样便有了目标函数：

$$\max \text{sum}(profit)$$

约束条件：

(1) 投资比例约束：每一年中，对某个行业的总投资比例大于 10%：

$$\frac{\text{industry} - \text{invest}_k}{\text{sum}(X)} \geq 0.01$$

值得强调的是 $x_{i,j}$ 表示第 i 年是否对项目 j 进行了投资，要计算项目 j 的总投资，需要对以往年份进行求和。

(2) 风险约束：题干给的约束条件是平均风险等级不低于 2.5。我们认为风险的大小和投资的本金数成正比例关系，所以在计算风险约束时，采用加权平均：

$$\sum_k \frac{\text{industry} - \text{invest}_k}{\text{sum}(X)} * f_k \geq 2.5$$

(3) 到期年限约束：平均到期年限不超过 8 年。同样，我们采用加权平均：

$$\sum_k \frac{industry_invest_k}{sum(X)} * y_k \leq 8$$

(4) 投资金额约束：第 i 年实际投资的资金不得超过当年本金，即为：

$$base_money_i - year_money_i \geq 0$$

(5) 本金约束：第 i 年可以用来投资的资金，有如下公式：

$$base_money_{i+1} = base_money_i + sum(profit_{t+1,j}) - sum(x_{i,j}) - student, 0 \leq j \leq 15$$

意思就是下一年可以用来投资的钱，等于，上一年的钱减去上一年投资，减去今年帮助贫困生的钱，加上今年收益。

综上所述，针对问题 1 我们可以建立如下模型：

$$\begin{aligned} & \max sum(Profit) \\ & s.t. \begin{cases} \frac{industry_invest_{i,k}}{sum(\{x_{i',j} \mid i' \leq i, j \leq 15\})} \geq 0.1 \\ \sum_k \frac{industry_invest_k}{sum(X)} * f_k \geq 2.5 \\ \sum_k \frac{industry_invest_k}{sum(X)} * y_k \leq 8 \\ base_money_i - year_money_i \geq 0 \\ x_{i,j} \geq 0 \\ base_money_{i+1} = \\ base_money_i + sum(profit_{i+1,j}) - sum(x_{i,j}) - student \\ (0 \leq j \leq 15) \end{cases} \end{aligned}$$

4.1.3 模型求解

我们用 python 中的 scipy 求得在 2021-2050 年债券的投资情况如表 2-表 6 所示，代码见附录 A。

表 2 2021-2026 年，各种债券的投资情况

	2021	2022	2023	2024	2025	2026
1	821.20	1936.00	0.00	0.00	1269.00	110.80
2	292.50	517.60	0.00	32.70	177.40	39.69
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

6	0.00	0.00	0.00	0.00	0.00	0.00
7	570.20	1195.00	0.00	333.80	84.33	0.00
8	175.40	175.40	175.40	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	5649.00	666.70	0.00	0.00	0.00	0.00

表 3 2027-2032 年，各种债券的投资情况

	2027	2028	2029	2030	2031	2032
1	853.80	610.80	0.00	495.40	426.60	232.80
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	223.70	0.00	0.00	129.30	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 4 2033-2038 年，各种债券的投资情况

	2033	2034	2035	2036	2037	2038
1	277.90	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00

12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 5 2039-2044 年，各种债券的投资情况

	2039	2040	2041	2042	2043	2044
1	0.00	1571.00	0.00	0.00	470.40	0.00
2	0.00	8000.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	568.70	0.00	0.00	568.70	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	1254.00	0.00	0.00	110.90	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	130.10	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	45.25	491.30	0.00	0.00	0.00

表 6 2045-2048 年，各种债券的投资情况

	2045	2046	2047	2048
1	350.90	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	568.70	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00

分析 1：最大收益为 14739 万元，可以看到，优化器选择了最后一种债券（15 号）作为第一年的最大投资对象，虽然该债券的税后年收益率不如 3 号债券，但是考虑到该债券的到期年限更低，更容易满足年限的约束，并且可以平衡该行业（国民福利）另外两种债券（13，14 号）的低收益率，从而更容易满足投资比例约束，所以优化器的选择是十分合理的。同时，模型也偏向于投资 1 号债券，这得益于 1 号债券拥有最高的风险级别和最短的到期年限的同时，其税后收益率也不算太低。

4.2 问题二

4.2.1 问题描述和分析

根据题干，基金会在 2021 年可以得到一笔不大于 5000 万的一次性贷款，每年利息 2%。也就是在问题一的基础上加入了如下条件：（1）2021 年初，本金增大；（2）每年从本金中扣除一部分，用以偿还利息。题干给出了各项投资的总收益率，可以根据公式：

$$(1 + p)^{y_j} = 1 + yield_j / 100$$

得到每项投资的年利率（税后）：

	风险	税率	到期年份	到期收益率	税后年收益率 %
医药健康	5	0	2	4.45	1.02201
	5	0	9	39.89	1.03800
	5	0	20	232.11	1.06185
交通运输	2	20	3	8	1.02089
	2	20	12	58.27	1.03240
	2	20	25	300.54	1.05022
科技研发	4	0	4	12.55	1.03000
	4	0	15	180.09	1.07107
	4	0	20	232.11	1.06185
装备制造	3	10	4	12.55	1.02711
	3	10	9	39.89	1.03467
	3	10	18	206.12	1.06001
国民福利	1	30	2	4.45	1.01546
	1	30	5	18.77	1.02500
	1	30	18	206.12	1.05087

从上表可以看出，只有国民福利领域的两年短期投资，它的税后收益率小于 2%；换句话说，只要我们将贷款进行投资，确保不投或者少投国民福利领域的两年短期投资，我们就一定能收益。再者因为我从银行获得的贷款是一次性贷款，每年利息不变，恒等于“贷款金额*利率”；如果我们将贷款 A 投资，本金将变大（增加 B），这样一来相对利率其实是在不断下降的：

$$\frac{A * 2\%}{A + B} < 2\%$$

综上所述，为了降低复杂度，我们直接从银行贷款 5000 万用于投资，每年偿还利息 100 万。

4.2.2 模型建立

基于上述分析，第 2 问将第 1 问约束 (5) 修改为：

(6) 本金约束：

$$base_money_{i+1} = base_money_i + sum(profit_{i+1,j}) - sum(x_{i,j}) - student - Interest$$

$$0 \leq j \leq 15$$

(7) $Base_money$ 初始值修改为：

$$Base_money = \{8000 + Loan, \dots, base_money_i, base_money_{i+1}, \dots\}$$

所以，针对问题 2 可以建立如下模型：

$$\begin{aligned} & \max \quad SUM(Profit) \\ & \left\{ \begin{array}{l} \frac{industry_invest_k}{sum(X)} \geq 0.1 \\ \sum_k \frac{industry_invest_k}{sum(X)} * f_k \geq 2.5 \\ \sum_k \frac{industry_invest_k}{sum(X)} * y_k \leq 8 \\ base_money_i - year_money_i \geq 0 \\ x_{i,j} \geq 0 \\ base_money_{i+1} = \\ base_money_i + sum(profit_{i+1,j}) - sum(x_{i,j}) - student - Interest \\ (0 \leq j \leq 15) \\ Interest = 100 \\ Loan = 5000 \end{array} \right. \end{aligned}$$

4.2.3 模型求解

我们用 python 中的 scipy 求得在 2021-2050 年债券的投资情况如表 7-表 11 所示，代码见附录 A。

表 7 2021-2026 年，各种债券的投资情况

	2021	2022	2023	2024	2025	2026
1	816.40	1521.82	0.00	40.66	777.47	777.47
2	1652.05	1136.28	0.00	0.00	0.00q	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00

4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	1412.30	2035.91	0.00	0.00	0.00	0.00
8	289.93	289.93	289.93	0.00	0.00	0.00
9	0.33	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00Q	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	8000	1102.00.	0.00	0.00	0.00	0.00

表 8 2027-2032 年，各种债券的投资情况

	2027	2028	2029	2030	2031	2032
1	1521.82	0.00	0.00	777.47	777.47	777.47
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	4.41	0.00	0.00	721.52	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0

表 9 2033-2038 年，各种债券的投资情况

	2033	2034	2035	2036	2037	2038
1	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00

10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 10 2039-2044 年，各种债券的投资情况

	2039	2040	2041	2042	2043	2044
1	0.00	3682.99	0.00	0.00	776.92	0.00
2	0.00	6952.50	28.61	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	1487.87	0.00	0.00	1860.04	473.45
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	801.69	0.00	0.00	397.85	0.00
11	0.00	3405.60	10.66	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	754.75	0.00	0.00	0.00

表 11 2045-2048 年，各种债券的投资情况

	2045	2046	2047	2048
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.00	795.01	0.00
5	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00

15	0.00	0.00	0.00	0.00
----	------	------	------	------

在得到银行 5000 万贷款之后，最大收益也达到了 21362 万元，并且对债券的投资比例也与第一问的结果差不太多，都选择了 15 号债券和 1 号债券作为主要投资对象。并且，第二问的投资相对于第一问更加分散，经过分析，这主要是为了每年偿还银行的利息，从而选择小比例的进行投资。

4.3 问题三

4.3.1 问题描述和分析

根据题干，在第二问的基础上，该题增加了若干个限制条件来约束投资比例。这些限制条件会打乱之前问题二中的投资方案，从而使得总的收益率变小，为了达到问题二中的收益金额，需要通过向银行继续贷款来增加本金，所以问题转换为最小化贷款金额问题。由于贷款金额不能直接进行优化，所以我们首先固定贷款金额来建立模型并计算出收益，然后通过调整贷款金额直至收益大于等于第二问中的收益。

4.3.2 模型建立

基于上述分析我们调整资金比例的约束，并将其分为四个约束：

(8) 医疗健康与科技研发投资之间的约束：

$$industry_invest_1 - industry_invest_3 * 90\% < 10^{-5}$$

(9) 医疗健康与装备制造投资之间的约束：

$$industry_invest_1 - industry_invest_4 * 190\% < 10^{-5}$$

(10) 医疗健康与交通运输投资之间的约束：

$$industry_invest_1 - industry_invest_2 * 290\% < 10^{-5}$$

(11) 医疗健康与国民福利投资之间的约束：

$$industry_invest_1 - industry_invest_5 * 390\% < 10^{-5}$$

(12) *Base _ money* 初始值修改为：

$$Base_money = \{8000 + Loan, \dots, base_money_i, base_money_{i+1}, \dots\}$$

综上所述，我们首先建立如下模型来表示贷款金额与收益的关系：

$$\begin{aligned}
& \max \text{Sum(Profit)} \\
& \left\{ \begin{aligned}
& \frac{\text{industry_invest}_k}{\text{sum}(X)} \geq 0.1 \\
& \sum_k \frac{\text{industry_invest}_k}{\text{sum}(X)} * f_k \geq 2.5 \\
& \sum_k \frac{\text{industry_invest}_k}{\text{sum}(X)} * y_k \leq 8 \\
& \text{base_money}_i - \text{year_money}_i \geq 0 \\
& x_{i,j} \geq 0 \\
& \text{base_money}_{i+1} = \\
& \text{base_money}_i + \text{sum}(\text{profit}_{i+1,j}) - \text{sum}(x_{i,j}) - \text{student} - \text{Interest} \\
& (0 \leq j \leq 15) \\
& \text{Interest} = \text{Loan} * 2\% \\
& \text{SUM(Profit)} \geq \text{SUM(Profit)}_in_problem_2 \\
& \text{industry_invest}_1 - \text{industry_invest}_3 * 90\% < 10^{-5} \\
& \text{industry_invest}_1 - \text{industry_invest}_4 * 190\% < 10^{-5} \\
& \text{industry_invest}_1 - \text{industry_invest}_2 * 290\% < 10^{-5} \\
& \text{industry_invest}_1 - \text{industry_invest}_5 * 390\% < 10^{-5}
\end{aligned} \right.
\end{aligned}$$

在求解过程中发现上述模型约束过于苛刻，优化器难以求解，于是将问题求解转换为如下方式：

为了缓解对题干中尽可能相等的约束，我们将约束转换为优化目标来求解，从而得到一个多目标优化问题，即对让收益尽可能的大，让投资量之间的差值尽可能的小。为了求解上述多目标优化问题，我们用线性加权法将其转换为单目标优化问题。

基于以上理解，我们建立如下模型：

$$\begin{aligned}
& \max \text{Sum}(\text{Profit}) + w * \sum_{k=1}^4 (\text{industry_invest}_0 - s_k * \text{industry_invest}_k) \\
& s.t. \begin{cases} \frac{\text{industry_invest}_k}{\text{sum}(X)} \geq 0.1 \\ \sum_k \frac{\text{industry_invest}_k}{\text{sum}(X)} * f_k \geq 2.5 \\ \sum_k \frac{\text{industry_invest}_k}{\text{sum}(X)} * y_k \leq 8 \\ \text{base_money}_i - \text{year_money}_i \geq 0 \\ x_{i,j} \geq 0 \\ \text{base_money}_{i+1} = \\ \text{base_money}_i + \text{sum}(\text{profit}_{i+1,j}) - \text{sum}(x_{i,j}) - \text{student_Interest} \\ (0 \leq j \leq 15) \\ \text{Interest} = \text{Loan} * 2\% \\ \text{SUM}(\text{Profit}) \geq \text{SUM}(\text{Profit})_{\text{in_problem_2}} \end{cases}
\end{aligned}$$

其中 s_k 表示医疗行业跟其他行业之间的倍数。

4.3.3 模型求解

表 12 2021-2026 年，各种债券的投资情况

	2021	2022	2023	2024	2025	2026
1	1121.44	1245.69	0.00	506.63	636.40	636.40
2	1756.49	930.10	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	1156.04	1227.21	0.00	0.00	0.00	0.00
8	2566.81	237.32	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	5734.49	664.72	0.00	0.00	0.00	0.00

表 13 2027-2032 年，各种债券的投资情况

	2027	2028	2029	2030	2031	2032
1	1245.69	0.00	0.00	636.40	636.40	636.40
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	1156.034	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 14 2033-2038 年，各种债券的投资情况

	2033	2034	2035	2036	2037	2038
1	0.00	0.00	0.00	636.40	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	6524.66	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 15 2039-2044 年，各种债券的投资情况

	2039	2040	2041	2042	2043	2044
1	0.00	5026.06	759.67	1186.64	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00

4	0.00	0.00	0.00	769.35	0.00	128.75
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	6597.31	0.00	3756.23	2375.20	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	966.56	2209.80	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00

表 16 2045-2048 年，各种债券的投资情况

	2045	2046	2047	2048
1	8000.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.00	4541.99	1030.73
5	0.00	0.00	0.00	1030.73
6	0.00	0.00	0.00	1030.73
7	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	3207.35
11	0.00	0.00	0.00	3207.35
12	0.00	0.00	0.00	3207.35
13	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00

由于时间关系，这里只求了一次解，假设贷了 5000 万的款，得到了 19941 万的收益，很明显达不到第二问中的捐助金额，但是此方法理论上可行，最后一定能得到贷款金额和捐助金额的关系，从而找到合适的解。

附录 A 程序代码

1.问题一

```
from scipy.optimize import minimize
import numpy as np
import matplotlib.pyplot as plt

# students_money = 200

np.set_printoptions(precision=3)
risk = [5, 5, 5, 2, 2, 2, 4, 4, 4, 3, 3, 3, 1, 1, 1]
taxrate = [0, 0, 0, 0.2, 0.2, 0.2, 0, 0, 0, 0.1, 0.1, 0.1, 0.3, 0.3, 0.3]
deadline = [2, 9, 20, 3, 12, 25, 4, 15, 20, 4, 9, 18, 2, 5, 18]
profitrate = [4.45, 39.89, 232.11, 8.0, 58.27, 300.54, 12.55, 180.09, 232.11, 12.55, 39.89, 206.12, 4.45, 18.77, 206.12]
# 优化目标函数

def func(x):
    x = np.reshape(x, [16, 30])
    x = x[:-1]
    Profit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j+ deadline[i]<30:
                Profit[i][j+ deadline[i]] = x[i][j]*(0.01*profitrate[i])*(1-taxrate[i])

    return -np.sum(Profit)
# 投资比例约束
def conv1(x):
    x = np.reshape(x, [16, 30])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    # 每个行业投资比例
    # print(industry_invest)
    # print(x)
    invest_div_sum = industry_invest/np.sum(x)

    return invest_div_sum-0.1
```

```

# 投资量约束
def conv2(x):
    x = np.reshape(x, [16, 30])
    x = x[:-1]

    store_money = x[-1]

    Profit = np.zeros([15, 30])
    adprofit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j + deadline[i] < 30:
                Profit[i][j+ deadline[i]] = x[i][j] *
(0.01 * profitrate[i]) * (1 - taxrate[i])
                adprofit[i][j+ deadline[i]] = x[i][j]
* (1+0.01 * profitrate[i]) * (1 - taxrate[i])
    profitsum = np.sum(Profit)
    students_money = profitsum/30
    base_money = np.zeros([30])
    # 每年的本金
    base_money[0] = 8000-students_money
    base_money[1] = store_money[0]-store_money[1]-students_money
    for i in range(1,base_money.shape[0]-1):
        base_money[i+1] = store_money[i]-students_money+np.sum(
adprofit[... ,i])-store_money[i+1]
        # base_money[i + 1] = np.sum(adprofit[... , i])
        # print(base_money[i])
    # 每年的投资
    year_invest = np.sum(x,axis=0) # 30
    return base_money - year_invest

# 年限约束
def conv3(x):
    x = np.reshape(x, [16, 30])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    date = 0
    for i in range(len(deadline)):

        date+= deadline[i]*area_invest[i]
    # 平均年限
    avgdata = date/np.sum(x)
    return 8-avgdata

```

```

# 风险约束
def conv4(x):
    x = np.reshape(x, [16, 30])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    risksum = 0
    for i in range(len(risk)):
        risksum+= risk[i]*area_invest[i]
    # 平均风险
    avgrisk = risksum/np.sum(x)
    return avgrisk-2.5

def f():
    # global students_money
    # for i in range(10):
    #     students_money=students_money-i*10

    con1 ={'type': 'ineq', 'fun': conv1}
    con2 ={'type': 'ineq', 'fun': conv2}
    con3 ={'type': 'ineq', 'fun': conv3}
    con4 ={'type': 'ineq', 'fun': conv4}

    con = [con1, con2, con3, con4]
    b = (0, 8000)
    bd = [b for i in range(480)]
    res = minimize(func, np.zeros([480])+10+1e-5, options={'maxiter':100000, 'disp':True}, constraints=con, bounds=bd)
    x = res['x']
    print(np.round(np.reshape(x, [16, 30]), 2))
    # print(store_money)

    print('max profit', func(x))
    print(conv1(x))
    print('本金-投资')
    print(conv2(x))
    print(conv3(x))
    print(np.round(conv4(x), 2))
    print(res['message'])
    Profit = np.zeros([15, 30])
    adprofit = np.zeros([15, 30])
    x = np.reshape(x, [16, 30])
    for i in range(Profit.shape[0]):

```

```

        for j in range(Profit.shape[1]):
            if j + deadline[i] < 30:
                Profit[i][j + deadline[i]] = x[i][j]
*   (0.01 * profitrate[i]) * (1 - taxrate[i])
                adprofit[i][j + deadline[i]] = x[i][j]
*   (1 + 0.01 * profitrate[i]) * (1 - taxrate[i])
        profitsum = np.sum(Profit)
        students_money = profitsum / 30
        base_money = np.zeros([30])
        # 每年的本金
        base_money[0] = 8000 - students_money
        base_money[1] = base_money[0] - students_money - np.sum(x[.
., 0])
        for i in range(1, base_money.shape[0] - 1):
            base_money[i+1] = base_money[i]-students_money-np.sum(x
[... ,i])+np.sum(adprofit[... ,i])
            # base_money[i + 1] = np.sum(adprofit[... , i])
            # print(base_money[i])
        # 每年的投资
        year_invest = np.sum(x, axis=0) # 30
        print(base_money)
        print(year_invest)

if __name__ == '__main__':
    f()

```

2.问题二

```

from scipy.optimize import minimize
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# students_money = 200

np.set_printoptions(precision=3, suppress=True)
risk = [5, 5, 5, 2, 2, 2, 4, 4, 4, 3, 3, 3, 1, 1, 1]
taxrate = [0, 0, 0, 0.2, 0.2, 0.2, 0, 0, 0, 0.1, 0.1, 0.1, 0.3, 0.3, 0.3]
deadline = [2, 9, 20, 3, 12, 25, 4, 15, 20, 4, 9, 18, 2, 5, 18]
profitrate = [4.45, 39.89, 232.11, 8.0, 58.27, 300.54, 12.55, 180.09, 232.11, 12.
55, 39.89, 206.12, 4.45, 18.77, 206.12]
# 优化目标函数

```

```

def func(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    Profit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j+ deadline[i]<30:
                Profit[i][j+ deadline[i]] = x[i][j]*(0.
01*profitrate[i])*(1-taxrate[i])

    return -np.sum(Profit)
# 投资比例约束
def conv1(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    # 每个行业投资比例
    # print(industry_invest)
    # print(x)
    invest_div_sum = industry_invest/np.sum(x)

    return invest_div_sum-0.1

# 投资量约束
def conv2(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]

    store_money = x[-1]

    Profit = np.zeros([15, 30])
    adprofit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j + deadline[i] < 30:
                Profit[i][j+ deadline[i]] = x[i][j] *
(0.01 * profitrate[i]) * (1 - taxrate[i])
                adprofit[i][j+ deadline[i]] = x[i][j]
* (1+0.01 * profitrate[i]) * (1 - taxrate[i])
    profitsum = np.sum(Profit)
    students_money = profitsum/30

```

```

    base_money = np.zeros([28])
    # 每年的本金
    base_money[0] = 13000-students_money-100
    base_money[1] = store_money[0]-store_money[1]-students_money-10
0
    for i in range(1,base_money.shape[0]-1):
        base_money[i+1] = store_money[i]-students_money+np.sum(
adprofit[...,i])-store_money[i+1]-100
        # base_money[i + 1] = np.sum(adprofit[..., i])
        # print(base_money[i])
    # 每年的投资
    year_invest = np.sum(x,axis=0) # 28
    return base_money - year_invest

# 年限约束
def conv3(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    date = 0
    for i in range(len(deadline)):

        date+= deadline[i]*area_invest[i]
    # 平均年限
    avgdata = date/np.sum(x)
    return 8-avgdata

# 风险约束
def conv4(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    risksum = 0
    for i in range(len(risk)):
        risksum+= risk[i]*area_invest[i]
    # 平均风险
    avgrisk = risksum/np.sum(x)
    return avgrisk-2.5

def f():
# global students_money
# for i in range(10):
#     students_money=students_money-i*10

```

```

con1 ={' type': 'ineq', ' fun': conv1}
con2 ={' type': 'ineq', ' fun': conv2}
con3 ={' type': 'ineq', ' fun': conv3}
con4 ={' type': 'ineq', ' fun': conv4}

con = [con1, con2, con3, con4]
b = (0, 8000)
bd = [b for i in range(448)]
res = minimize(func, np.zeros([448])+10+1e-5, options={'maxiter':100000, 'disp':True}, constraints=con, bounds=bd)
x = res['x']
print(np.round(np.reshape(x, [16, 28]), 2))
# print(store_money)

print('max profit', func(x))
print(conv1(x))
print('本金-投资')
print(conv2(x))
print(conv3(x))
print(np.round(conv4(x), 2))
print(res['message'])

x = x[:-1]
a2=pd.DataFrame(x)
a2.to_excel('data.xlsx', index=False) #不包括行索引

if __name__ == '__main__':
    f()

```

3.问题三

```

from scipy.optimize import minimize
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
borrow_money = 5000
# students_money = 200

np.set_printoptions(precision=3)
risk = [5, 5, 5, 2, 2, 2, 4, 4, 4, 3, 3, 3, 1, 1, 1]
taxrate = [0, 0, 0, 0.2, 0.2, 0.2, 0, 0, 0, 0.1, 0.1, 0.1, 0.3, 0.3, 0.3]

```

```

deadline = [2, 9, 20, 3, 12, 25, 4, 15, 20, 4, 9, 18, 2, 5, 18]
profitrate = [4.45, 39.89, 232.11, 8.0, 58.27, 300.54, 12.55, 180.09, 232.11, 12.55, 39.89, 206.12, 4.45, 18.77, 206.12]
# 优化目标函数

def func(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    Profit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j+ deadline[i]<30:
                Profit[i][j+ deadline[i]] = x[i][j]*(0.01*profitrate[i])*(1-taxrate[i])
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    return -np.sum(Profit)+10*((industry_invest[0]-3*industry_invest[1])**2+(industry_invest[0]-industry_invest[2])**2+(industry_invest[0]-2*industry_invest[3])**2+(industry_invest[0]-4*industry_invest[4])**2)

# 投资比例约束 1
def conv1_1(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    return industry_invest[0]-3*industry_invest[1]

# 投资比例约束 2
def conv1_2(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    return industry_invest[0]-1*industry_invest[1]

# 投资比例约束 3
def conv1_3(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    return industry_invest[0]-2*industry_invest[1]

```

```

# 投资比例约束 4
def conv1_4(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1) # 5
    return industry_invest[0]-4*industry_invest[1]

# 投资量约束
def conv2(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]

    store_money = x[-1]

    Profit = np.zeros([15, 30])
    adprofit = np.zeros([15, 30])
    for i in range(Profit.shape[0]):
        for j in range(Profit.shape[1]):
            if j + deadline[i] < 30:
                Profit[i][j+ deadline[i]] = x[i][j] *
(0.01 * profitrate[i]) * (1 - taxrate[i])
                adprofit[i][j+ deadline[i]] = x[i][j]
* (1+0.01 * profitrate[i]) * (1 - taxrate[i])
    profitsum = np.sum(Profit)
    students_money = profitsum/30
    base_money = np.zeros([28])
    # 每年的本金
    base_money[0] = 8000-students_money+borrow_money
    base_money[1] = store_money[0]-store_money[1]-students_money-bo
rrow_money*0.02
    for i in range(1, base_money.shape[0]-1):
        base_money[i+1] = store_money[i]-students_money+np.sum(
adprofit[... , i])-store_money[i+1]
        # base_money[i + 1] = np.sum(adprofit[... , i])
        # print(base_money[i])
    # 每年的投资
    year_invest = np.sum(x, axis=0) # 30
    return base_money - year_invest

# 年限约束
def conv3(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]

```

```

    area_invest = np.sum(x, axis=1) # 15
    date = 0
    for i in range(len(deadline)):

        date+= deadline[i]*area_invest[i]
    # 平均年限
    avgdata = date/np.sum(x)
    return 8-avgdata

# 风险约束
def conv4(x):
    x = np.reshape(x, [16, 28])
    x = x[:-1]
    area_invest = np.sum(x, axis=1) # 15
    risksum = 0
    for i in range(len(risk)):
        risksum+= risk[i]*area_invest[i]
    # 平均风险
    avgrisk = risksum/np.sum(x)
    return avgrisk-2.5

def f():
# global students_money
# for i in range(10):
#     students_money=students_money-i*10

    con11 ={'type': 'eq', 'fun': conv1_1}
    con12 ={'type': 'eq', 'fun': conv1_2}
    con13 ={'type': 'eq', 'fun': conv1_3}
    con14 ={'type': 'eq', 'fun': conv1_4}
    con2 ={'type': 'ineq', 'fun': conv2}
    con3 ={'type': 'ineq', 'fun': conv3}
    con4 ={'type': 'ineq', 'fun': conv4}

    con = [con2, con3, con4]
    b = (0, 8000)
    bd = [b for i in range(448)]
    res = minimize(func, np.zeros([448])+10+1e-5, options={'maxiter':100000,'disp':True}, constraints=con, bounds=bd)
    x = res['x']
    print(np.round(np.reshape(x, [16, 28]), 2))
    # print(store_money)

```

```

print('max profit', func(x))
print(conv1_1(x))
print(conv1_2(x))
print(conv1_3(x))
print(conv1_4(x))
print('本金-投资')
print(conv2(x))
print(conv3(x))
print(np.round(conv4(x), 2))
print(res['message'])
x = np.reshape(x, [16, 28])
x = x[:-1]
a2=pd.DataFrame(x)
a2.to_excel('data1.xlsx', index=False)    #不包括行索引

Profit = np.zeros([15, 30])
for i in range(Profit.shape[0]):
    for j in range(Profit.shape[1]):
        if j+ deadline[i]<30:
            Profit[i][j+ deadline[i]] = x[i][j]*(0.
01*profitrate[i])*(1-taxrate[i])
    area_invest = np.sum(x, axis=1)    # 15
    industry_invest = np.sum(area_invest.reshape(5, 3), axis=1)
# 5
    print(-np.sum(Profit))
    print(industry_invest[0])
    print(industry_invest[1]*3)
    print(industry_invest[2])
    print(industry_invest[3]*2)
    print(industry_invest[4]*4)
if __name__ == '__main__':
    f()

```