

Задача.

Входные данные:

- Матрица **S** размером $N_f \times N_a$, заполненная большей частью нулями.
- Вектор **a**, $\vec{a} = (a_1, \dots, a_{N_a})$:
- Вектор **b**, $\vec{b} = (b_1, \dots, b_{N_f})$:
- Вектор **f**, $\vec{f} = (f_1, \dots, f_{N_f})$

Найти: матрицу **D**, удовлетворяющую условиям:

1. $d_{ij} = 0$, если $s_{ij} = 0, (1 \leq i \leq N_f, 1 \leq j \leq N_a)$
т.е. в матрице **D** не меньше нулей, чем в **S**
2. $\sum_{i=1}^{N_f} d_{ij} \leq a_j, (1 \leq j \leq N_a)$
т.е. суммы элементов в столбцах матрицы **D** ограничены компонентами вектора **a**
3. $\sum_{j=1}^{N_a} d_{ij} \leq b_i, (1 \leq i \leq N_f)$
т.е. суммы элементов в строках матрицы **D** ограничены компонентами вектора **b**
4. $\sum_i \sum_j d_{ij} \leq A$
т.е. сумма всех элементов ограничена суммой компонентов вектора **a**
5. $\frac{d_{ij}}{s_{ij}} = \frac{d_{ik}}{s_{ik}}, \forall j, k$
строки матриц **S** и **D** подобны, при этом в **D** могут быть нулевые строки.
6. $R = \max$

Обозначения:

$$A = \sum_{i=1}^{N_a} a_i$$

$$R = \sum_{i=1}^{N_f} f_i \sum_{j=1}^{N_a} d_{ij} + (A - \sum_i \sum_j d_{ij})$$

Т.е. из всех возможных вариантов матрицы **D**, удовлетворяющим условиям (1-5), необходимо отыскать такую, при которой **R** будет максимальным (6).

Этапы выполнения задания

1. Создание алгоритма решения (это должен быть не тупой перебор всех возможных комбинаций, а именно алгоритм).
2. Генерация тестовых заданий (матрицы и векторов)
3. Проверка решения.

Это означает, что в ходе работы над проектом должны быть созданы следующие консольные приложения на C++ :

1. `Generator.exe Nf Na`, где Nf, Na – аргументы командной строки. Возможно надо будет добавить другие аргументы для генерации, например, пределы изменения значений.

2. Solver.exe *S.txt a.txt b.txt f.txt* – должна формировать итоговый файл *D.txt* и выводить на экран значения *A* и *R*.
3. Checker.exe *a.txt D.txt* – должна выводить на экран значения *A* и *R*.

Для файлов, содержащих векторы, сначала считывается размер N , затем N его элементов.
Для файлов, содержащих матрицы, сначала считываются размеры N_f , N_a , а затем $N_f \times N_a$ его элементов.

Код для Generator и Checker практически готов (но на джаве), могу выслать, хотя эти программы несложно реализуются.

Условия работы:

- 1) вы разрабатываете свой алгоритм,
- 2) пишете программы,
- 3) отправляете их без исходников,
- 4) я тестирую,
- 5.1) если нахожу, что в ходе проверки мною найдено более оптимальное решение (с бо́льшим R), то я вам высылаю файлы с этим решением, после чего вам необходимо выяснить, в чем дело,
- 5.1.1) если вы не укладываетесь в сроки, то я вправе прекратить сотрудничество
- 5.2) если всё в порядке, я оплачиваю,
- б) вы присылаете исходники.

Всё общение через сайт [free-lance.ru](https://www.free-lance.ru).