

Как я перестал бояться и полюбил LP-солверы

В некоторых задачах вам потребуется написать решения, использующие сведение к задаче линейного или смешанного программирования. Вам доступны следующие опции: вы можете либо сами написать любой доступный вам алгоритм, либо воспользоваться одной из двух выбранных нами библиотек для языков C/C++ и Python2 (к сожалению, Python 3 в этот раз доступен не будет).

GLPK

GLPK — промышленная open-source библиотека, предоставляющая API для языков C и C++, и содержащая ряд процедур для численного решения задач различных общих форм, в том числе, задач линейного программирования.

Чтобы воспользоваться данной библиотекой в своём решении на языке C или C++, нужно подключить заголовочный файл `"glpk.h"` (именно так, в двойных кавычках), и послать решение в тестирующую систему с использованием компилятора Make2 и расширением файла `.c` или `.cpp`.

Полная документация по данной библиотеке доступна в виде pdf-документа. Рекомендуем внимательно её проглядеть, чтобы не упустить никакую из мощных фишек данной библиотеки.

Чтобы скомпилировать решение с использованием данной библиотеки локально под Linux, надо откуда-нибудь раздобыть заголовочный файл `glpk.h` и библиотеку `libglpk.so/libglpk.a`. Самый простой способ сделать это в debian-подобных системах — поставить пакет `libglpk-dev` командой `sudo apt-get install libglpk-dev`. Можно также собрать пакет из исходников, которые можно скачать здесь, командой `./configure && make`, и унести из него файлы `src/glpk.h` и `src/.libs/libglpk.a` в директорию со своей программой.

Для того, чтобы скомпилировать своё решение, можно воспользоваться командой `g++ <name>.cpp -o <name> -lglpk [-L.]` (последний ключ нужен, если вы положили библиотеку `libglpk.a` локально).

Пара замечаний по пользованию библиотекой. Во-первых, обратите внимание, что все массивы, которые вы будете передать в процедуры, предоставляемые библиотекой, индексируются с единицы (в частности, нулевой элемент не имеет значения). Во-вторых, по умолчанию библиотека выводит много отладочной информации в стандартный вывод; если решение с её использованием отправить на проверку в тестирующую систему, то вы, скорее всего, получите вердикт «Неправильный формат вывода». Отключите отладочный вывод путём передачи в процедуру симплекс-алгоритма параметра `msg_lev`, равного `GLP_MSG_OFF`.

SciPy

SciPy — очень мощный фреймворк для Python, предоставляющий множество различных примитивов, и в том числе позволяющая решать задачи линейного программирования (но, к сожалению, без задач смешанного программирования).

Документация по нужной части фреймворка доступна [здесь](#).

Для того, чтобы воспользоваться данной библиотекой, нужно также послать решение с использованием компилятора Make2 и расширением `.py`.

Для локального тестирования удобнее всего поставить данную библиотеку с помощью менеджера пакетов Python `pip`. В debian-подобных системах `pip` устанавливается командой `sudo apt-get install python-pip`, после чего можно установить SciPy командой `sudo -H pip install scipy`. По итогам запуска этой команды в интерпретаторе Python должна работать команда `import scipy`.