

Домашнее задание 3 по дискретной оптимизации. Задача о покраске

Кондратюк В., Хильдебранд Р.

Дисклеймер

В каждой домашке предлагается написать алгоритм решения для задачи оптимизации. Берите во внимание тот факт, что постановка задачи, предложенная в условии будет максимально сырой и неудобной, так что подумайте как переформулировать задачу. Вместе с задачей мы предложим набор тестов, из которых 6 будут оцениваться.

Решением задания являются 3 файла: код на любом языке программирования (укажите, как его собрать и запустить, желательно напишите всю сборку баш-скриптом и упростите жизнь проверяющим), pdf с описанием решения и чекер, который запускает Ваш код на шести наших тестовых наборах, проверяет правильность решения и выводит значение целевой функции. Помимо описания метода нужно предоставить отчёт о хотя бы трёх улучшениях, которые Вы попробовали (даже если качество не было лучшим) и привести анализ/обоснование выбранных методов. Рекомендуется решать задачу итеративно: написать какое-то решение, протестировать его на всех тестах, поискать аномалии в решении и начать их исключать.

Сдачей решения является один код, получать ответы к разным тестовым данным нужно при помощи одной программы. Однако, в ней могут быть if-ы по размеру входа, если Вы обоснуете их правильность.

Оценивается каждое задание из 50 баллов, которые получаются следующим образом: для каждого теста будет два целевых значения, которые мы Вам сообщим. Получение хоть какого-нибудь нетривиального ответа даст 3 балла за тест, пробитие первого порога — 5 баллов, достижение третьего порога — 7 (третий порог непробиваем, так как является глобальным оптимумом), 8 баллов стоит отчёт, но если у Вас меньше 25 баллов за тесты, то за отчёт Вы получите не более 4.

Использовать сторонние солверы запрещено!

Работать код должен до получаса на каждом teste, иначе он не оценивается.

Старайтесь не подгонять солвер под данные и напишите в отчёте, почему Ваш подход работает в каких-то ожидаемых случаях. Явный подгон под структуру тестов (и, в худшем случае, if с выведением скатанного ответа) будет караться пересдачей за курс.

Делитесь идеями в чате, задавайте вопросы. Запрещается только показывать друг другу код, но можно пересказывать его принцип работы в виде сказки.

Тихону Евтееву разрешается всё то же, что и остальным.

Условие

Задача о покраске графа формулируется следующим образом:

$$\min \left[\max_{i=0 \dots n-1} c_i \right], \text{ s.t. } \text{alldifferent}(\{c_i, i \in E\}), \quad (1)$$

где E — двухэлементные множества (ребра графа), $\text{alldifferent}(A)$ означает, что все элементы списка A различны.

Рекомендуется попробовать какие-то эвристики промоделировать на языке MiniZinc. Прикладывайте к отчёту код на этом языке, при помощи которого Вы поняли, что какая-то эвристика хорошо работает. За эксперименты с MiniZinc мы будем добавлять 1 балл в отчёт (при максимуме 5).

Формат входа

В файле с тестом в первой строке находятся два натуральных числа: n — число вершин, m — число ребер. Вершины нумеруются с нуля.

В следующих m строках находятся пары натуральных чисел: x_i, y_i — номера вершин ребра.

По адресу <https://github.com/discreteoptimization/visualization/tree/master/coloring> лежит визуализатор Вашей покраски.

Тесты и пороги

В списке ниже приведены названия файлов тестов, которые мы оцениваем и пороги на 5 и 7 баллов:

- gc_50_3, 8, 6
- gc_70_7, 20, 17
- gc_100_5, 21, 16
- gc_250_9, 95, 78
- gc_500_1, 18, 16
- gc_1000_5, 124, 100

Дедлайн

Чтобы не получить обнуление домашки, нужно к 3 апреля 23:59 предоставить код, который способен найти нетривиальное решение с любой целевой функцией для хотя бы трёх тестов с описанием метода, а к 17 апреля представить итоговое решение. Куда его сдавать будет сообщено позже. Да, мы поднимем ахчек.