

# Комбинаторная оптимизация

## Car sequencing

9 июня 2019 г.

### 1 Формулировка

Вы занимаетесь задачей оптимизации производства автомобилей. Ваша компания считается прогрессивной за счет того, что она позволяет потребителям сконфигурировать свой уникальный автомобиль путем добавления различных *опций* к базовой конфигурации. Тем не менее, такая возможность сильно усложняет процесс производства автомобилей, оптимизацией которого вам и предстоит заняться.

Автомобили собираются по очереди. Очередь из автомобилей на сборку называется *линией сборки*, она представляет из себя последовательность из  $n$  целых чисел, задающих типы автомобилей в порядке сборки. Завод производит  $k$  различных типов автомобилей. Каждый тип описывается набором опций, которые необходимо установить на автомобили данного типа.

Опции добавляются на автомобили специальными роботами,  $i$ -й робот отвечает за установку  $i$ -й опции и описывается двумя целыми числами  $w_{i, cap}$  и  $w_{i, size}$ . Для  $i$ -го робота должно выполняться следующее ограничение: среди любых  $w_{i, size}$  последовательных автомобилей на линии сборки не более  $w_{i, cap}$  из них должны требовать  $i$ -ю опцию, иначе робот не будет успевать делать свою работу.

В идеале хочется расположить машины на линии сборки в таком порядке, что для всех роботов будет выполнено указанное ограничение. Однако поиск подобной конфигурации является сложной задачей, поэтому вам предстоит заняться поиском приближённого решения. Ваша задача заключается в поиске решения с минимальной суммарной величиной нарушения.

Величиной нарушения робота  $i$  на окне из  $w_{i, size}$  подряд идущих машин, заканчивающейся на  $j$ -й машине ( $0 \leq j \leq n + w_{i, size} - 2$ ), называется величина  $\max\{0, w_{i, req} - w_{i, cap}\}$ , где  $w_{i, req}$  — это количество машин в окне, требующих  $i$ -ю опцию (таким образом, величина нарушения равна нулю тогда и только тогда, когда на окне выполнено требуемое условие). Обратите внимание, окно может содержать и менее, чем  $w_{i, size}$  машин, если оно выходит за границы линии сборки слева или справа.

Суммарной величиной нарушения называется сумма величин нарушения по всем роботам и по всем окнам, соответствующим каждому роботу (в том числе по окнам, которые выходят за границы сборки).

## 2 Входные данные

Первая строка входных данных содержит три целых числа  $n$  ( $1 \leq n \leq 200$ ),  $l$  ( $1 \leq l \leq 2000$ ) и  $k$  ( $1 \leq k \leq 100$ ) — количество машин в линии сборки, количество различных опций и количество различных типов машин. Указанные ограничения верны для всех закрытых тестов, открытые тесты могут выходить за данные ограничения.

Следующая строка содержит  $l$  чисел  $w_{i, cap}$  ( $1 \leq w_{i, cap} \leq n$ ), описывающих максимально возможное допустимое число обрабатываемых автомобилей на окне для каждого из роботов.

Следующая строка содержит  $l$  чисел  $w_{i, size}$  ( $w_{i, cap} \leq w_{i, size} \leq n$ ), описывающих размеры окон роботов.

В следующих  $k$  строках описаны типы автомобилей. Типы автомобилей обозначаются целыми числами от 0 до  $k - 1$ , описания типов даны в порядке от 0 до  $k - 1$ . Описание начинается с целого положительного числа  $n_i$  ( $\sum_{i=0}^{k-1} n_i = n$ ), задающего количество автомобилей типа  $i$ , которые надо изготовить. Далее на той же строке следуют  $l$  чисел из множества  $\{0, 1\}$ , описывающих наличие или отсутствие соответствующей опции у данного типа автомобилей.

## 3 Формат вывода

В первой строке выведите число  $v$  — суммарную величину нарушения в вашем решении.

Во второй строке выведите  $n$  целых чисел от 0 до  $k - 1$  — номера типов автомобилей в порядке следования вдоль линии сборки в вашем решении.

## 4 Оценивание

Для каждого теста определена пороговая суммарная величина нарушения, вам неизвестная. Если найденный вами ответ не превосходит данной пороговой величины, то тест считается пройденным. В тестирующей системе доступно 13 тестов, из которых первые 5 являются открытыми и доступны вам в виде архива для самостоятельного тестирования, а последние 8 являются закрытыми. Каждый закрытый тест оценивается в 0.5 балла.

Архив с открытыми тестами состоит из 5 файлов без расширения непосредственно с тестами, 5 файлов, имеющих расширение `.a`, в которых находятся пороговые значения для соответствующих тестов, а также 5 файлов, имеющих расширение `.r`, в которых находятся примеры решений, допускаемых пороговыми значениями.

Также необходимо предоставить отчёт, в котором должна быть ссылка на посылку в Яндекс.Контесте, описание идей и экспериментов. Отчет оценивается из 0.5 балла.

Если ваш ответ будет некорректным (неправильно посчитанная суммарная величина нарушения, неправильное количество автомобилей нужного вида), вы получите вердикт «Неправильный формат вывода» на соответствующем тесте. Если ваш ответ корректен, но доставляет значение выше порогового, вы получите вердикт «Неправильный ответ» на соответствующем тесте.

## 5 Полезные советы

Попробуйте применить методы локального поиска. Для этого стоит ответить на вопросы:

1. Какие ограничения задачи будут нарушаться при локальном поиске, а какие всегда будут выполнены?
2. Какая система соседства будет гарантировать инвариант из предыдущего пункта?
3. Какое правило выбора соседа будет использоваться? (Select function).
4. Как бороться с тем, что решение застревает в локальных минимумах?