Кўриш дастури

Сиз робот учун кўриш дастурини тайёрламоқчисиз.

Робот хар сафар тасвирни олганда, у роботнинг хотирасида оқ-қора расм сифатида сақланади. Хар бир тасвир $H \times W$ ўлчамли тўғри бурчакли сетка кўринишида бўлади, сатрлар 0 дан $H-1$ гача, устунлар 0 дан $W-1$ гача номерланади (рақамланади).

Тасвирда **аниқ иккита** қора пиксель бор, қолган барча пикселлар оқ.

Робот содда инструкциялардан иборат дастурни қўллаб хар бир тасвирни қайта ишлаб чиқиши мумкин.

Сизга H и W сонлар ва K. мусбат сон берилган.

Сизнинг мақсадингиз робот учун программа ёзадиган шундай функцияни топиш керакки, ушбу программа хар бир тасвир учун қора пикселлар орасидаги **масофа** K га аниқ тенглигини аниқласин.

Бу холда r_1 сатр ва c_1 устундаги пиксель ҳамда r_2 сатр ва c_2 устундаги пиксель орасидаги масофа $|r_1 - r_2| + |c_1 - c_2|$ га тенг.

Роботнинг тузилишини таърифлаймиз. Роботнинг хотираси катакларнинг катта массиви бўлиб, 0 дан бошлаб индексланади.

Хар бир катак ё 0 ни ёки 1 ни сақлайди, бу маълумот ўрнатилганидан сўнг ўзгармайди. Тасвир сатрлар бўйлаб катакларда 0 индексдан $H\,cdotW-1$ индексгача сақланади.

Биринчи сатр номерлари 0 дан $W-1$ гача бўлган катакчаларда, охирги сатр номерлари $(H-1)\,cdotW$ дан $H \cdot W - 1$ гача бўлган катакчаларда сақланади.

Хусусан, i сатр ва j устундаги пиксель қора бўлса, $i \cdot W + j$ катакча қиймати 1 га тенг, акс холда 0 га тенг.

Роботнинг дастури **инструкциялар - яъни кўрсатмалар** кетма-кетлиги бўлиб, 0 дан бошлаб кетма-кет рақамлар билан рақамланган.

Дастур ижроси давомида ушбу кўрсатмалар навбат билан бажарилади.

Хар бир кўрсатма бир ёки бир нечта катакнинг қийматини ўқийди (уларни кўрсатмаларнинг **кириш қийматлари** деб атаймиз) ва ва чиқишда 0 ёки 1 га тенг

ягона сон беради (бу қийматларни кўрсатмаларнинг **натижалари** деб атаймиз).

i кўрсатманинг натижаси $H \cdot W + i$ катакда бўлади. i кўрсатманинг кириш маълумотлари ёки пиксельлар, ёки олдинги кўрсатмаларнинг натижаларини сақлаётган катак бўлиши мумкин, яъни 0 дан $H \cdot W + i - 1$ гача катакчалар.

Тўрт хил турдаги кўрсатмалар бўлиши мумкин:

- `NOT`: аниқ битта кириш қийматига эга. Кириш қиймати 0 га тенг бўлса, натижа 1 га тенг, акс холда 0 га тенг.
- `AND`: битта ёки ундан ортиқ кириш қийматига эга. Фақат ва фақат **барча** кириш қийматлари 1 га тенг бўлганда, натижа 1 га тенг.
- `OR`: битта ёки ундан ортиқ кириш қийматига эга. Фақат ва фақат **хеч бўлмаганда** битта кириш қиймати 1 га тенг бўлганда натижа 1 га тенг.
- `XOR`: битта ёки ундан ортиқ кириш қийматига эга. Фақат ва фақат **тоқ сонли** кириш қиймати 1 га тенг бўлганда, натижа 1 га тенг.

Агар иккита оқ пикселлар орасидаги масофа K бўлса, дастурнинг охирги кўрсатмаси ижроси натижаси 1 га тенг бўлиши, акс холда 0 га.

Детали реализации

Вам необходимо реализовать следущую функцию:

- H, W: размерность каждого изображения, снятого камерой робота.
- K: положительное целое число.
- Функция должна произвести программу для робота. Для каждого изображения, снятого камерой робота, программа должна определить, равно ли расстояние между двумя черными пикселями ровно K.

Эта функция должна вызывать одну или более следующих функций для добавления инструкций в программу робота (которая изначально является пустой):

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- Добавить инструкции `NOT`, `AND`, `OR`, или `XOR` соответственно.
- N (для `add_not`): индекс клетки, откуда добавленная инструкция `NOT` будет считывать входные данные.
- Ns (для `add_and`, `add_or`, `add_xor`): массив, содержащий индексы клеток, откуда `AND`, `OR`, or `XOR` считывают входные данные.
- Каждая функция возвращает индекс клетки, хранящей результат выполнения

инструкции. Последовательные вызовы этих функций возвращают последовательные целые числа, начинающиеся от $H \cdot W$.

Программа робота может содержать не более $10\,000$ инструкций. Инструкции могут считать суммарно не более $1\,000\,000$ значений. Другими словами, суммарная длина массивов Ns во всех вызовах `add_and`, `add_or` и `add_xor` плюс число вызовов `add_not` не может превышать $1\,000\,000$.

После добавления последней инструкции, функция `construct_network` должна завершиться.

Затем, программа робота будет оценена на определенном количестве изображений.

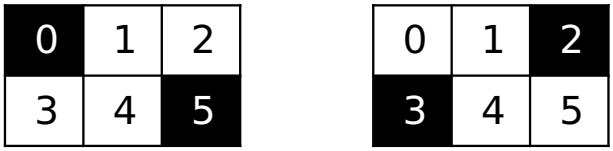
Ваше решение считается прошедшим определенным тест если для каждого из изображений в нём последняя инструкция возвращает 1 тогда и только тогда, когда расстояние между двумя пикселями изображения равно K.

В процессе оценивания Вашего решения может возникнуть одно из сообщений об ошибке, описанные ниже:

- `Instruction with no inputs`: был передан пустой массив функциям `add_and`, `add_or`, или `add_xor`.
- `Invalid index`: неверный (возможно, отрицательный) индекс клетки был передан функциям `add_and`, `add_or`, `add_xor`, или `add_not`.
- `Too many instructions`: Ваша функция попыталась добавить более чем $10\,000$ инструкций.
- `Too many inputs`: инструкции считали суммарно более чем $1\,000\,000$ значений.

Пример

Пусть $H = 2$, $W = 3$, $K = 3$. Существует всего два возможных изображения с расстоянием между двумя черными пикселями, равным 3.



- Вариант 1: черными пикселями являются 0 и 5
- Вариант 2: черными пикселями являются 2 и 3

Возможное решение заключается в том, чтобы составить программу робота со следующими инструкциями:

1. `add_and([0, 5])`, которая добавляет инструкцию, возвращающую 1 тогда и только тогда, когда выполняется Вариант 1. Результат сохраняется в клетке 6.
2. `add_and([2, 3])`, которая добавляет инструкцию, возвращающую 1 тогда и только тогда, когда выполняется Вариант 2. Результат сохраняется в клетке 7.
3. `add_or([6, 7])`, которая добавляет инструкцию, возвращающую 1 тогда и только тогда, когда выполняется один из вышеуказанных вариантов.

Ограничения

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

Подзадачи

1. (10 баллов) $\max(H, W) \leq 3$
2. (11 баллов) $\max(H, W) \leq 10$
3. (11 баллов) $\max(H, W) \leq 30$
4. (15 баллов) $\max(H, W) \leq 100$
5. (12 баллов) $\min(H, W) = 1$
6. (8 баллов) Пиксель в строке 0 и столбце 0 черный во всех изображениях
7. (14 баллов) $K = 1$
8. (19 баллов) Никаких дополнительных ограничений.

Пример проверяющего модуля

Пример проверяющего модуля считывает входные данные в следующем формате:

- строка 1: H W K
- строка $2 + i$ $(i \geq 0)$: r_1 c_1 r_2 c_2
- последняя строка: -1

Каждая строка за исключением первой и последней представляет собой изображения с двумя черными пикселями.

Мы называем изображение, описанное на строке $2 + i$ изображением i. Один черный пиксель находится в строке r_1 и столбце c_1 и другой - в строке r_2 и столбце c_2. Пример проверяющего модуля может вывести сообщение `Invalid user input` если обнаружатся ошибки во входных данных (т.е. входные данные содержат несуществующую строку или столбец).

Если ошибок не было найдено, пример проверяющего модуля выведет на экран результат работы программы работа в следующем формате:

- строка $1 + i$ $(0 \leq i)$: результат последней инструкции в программе робота для изображения i (1 для 0). В добавок к этому, пример проверяющего модуля пишет в файл `log.txt` в текущей директории в следующем формате:
- строка $1 + i$ $(0 \leq i)$: $m[0]$ $m[1]$ $\ldots$ $m[c-1]$

Последовательность в строке с номером $1 + i$ описывает значения, которые хранятся в памяти робота после выполнения программы робота, получившей изображение i в качестве входного изображения.

Более точно, $m[j]$ задачет значение клетки j. Обратите внимание, что значение c (длина последовательности) равно $H \cdot W$ плюс число инструкций в программе робота.