

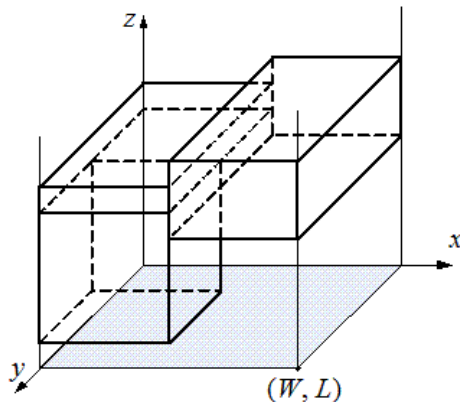
Яндекс. Тренировки по алгоритмам июнь 2021, занятие 7

J. НГУ-стройка

	Все языки	Python 3.6
Ограничение времени	2 секунды	4 секунды
Ограничение памяти	64Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Над ареной огромного спортивного комплекса Независимого Главного Университета (НГУ) решили построить перекрытие. Перекрытие будет построено по клеевой технологии и состоять из склеенных друг с другом блоков. Блок представляет собой легкий прямоугольный параллелепипед. Два блока можно склеить, если они соприкасаются перекрывающимися частями боковых граней ненулевой площади. НГУ представил план комплекса, имеющий вид прямоугольника размером W на L . При этом один из углов прямоугольника находится в начале системы координат, а другой имеет координаты (W, L) . Стены комплекса параллельны осям координат.

Подрядчики известили НГУ, что они готовы к определенному сроку изготовить блоки и установить их. Для каждого блока фиксировано место его возможного монтажа, совпадающее по размерам с этим блоком. Места выбраны так, что ребра блоков параллельны осям координат. Места монтажа блоков не пересекаются.



По техническим условиям перекрытие должно состоять из такого набора склеенных блоков, который содержит сплошной горизонтальный слой ненулевой толщины. Торопясь ввести комплекс в эксплуатацию, НГУ решил построить перекрытие из минимально возможного числа блоков.

Требуется написать программу, которая позволяет выбрать минимальное число блоков, которые, будучи установленными на указанных подрядчиками местах, образуют перекрытие, либо определить, что этого сделать невозможно. Высота, на которой образуется перекрытие, не имеет значения.

Формат ввода

В первой строке входного файла указаны три целых числа: N — количество возможных блоков ($1 \leq N \leq 10^5$) и размеры комплекса W и L ($1 \leq W, L \leq 10^4$). Каждая из последующих N строк описывает место монтажа одного блока, определяемое координатами противоположных углов: (x_1, y_1, z_1) и (x_2, y_2, z_2) , при этом $0 \leq x_1 < x_2 \leq W$, $0 \leq y_1 < y_2 \leq L$, $0 \leq z_1 < z_2 \leq 10^9$. Все числа во входном файле целые и разделяются пробелами или переводами строк. Гарантируется, что места установки блоков не пересекаются друг с другом.

Формат вывода

Первая строка выходного файла должна содержать либо слово «YES», если перекрытие возможно построить, иначе — слово «NO». В первом случае вторая строка выходного файла должна содержать минимальное число блоков, образующих перекрытие, а последующие строки — номера этих блоков, в соответствии с порядком, в котором они перечислены во входном файле. Если возможно несколько минимальных наборов блоков, выведите любой из них.

Пример 1

Ввод

```
1 10 10
0 0 0 10 10 10
```

Вывод

```
YES
1
1
```

Пример 2

Ввод

```
2 10 10
0 0 0 10 5 5
0 5 5 10 10 10
```

Вывод

```
NO
```

Язык

[Набрать здесь](#)

[Отправить файл](#)

```
1 # считываем данные
2 N, W, L = map(int, input().split())
3 total_area = W * L
4 events = []
5 for i in range(1, N + 1):
6     x1, y1, z1, x2, y2, z2 = map(int, input().split())
7     area = abs(x2 - x1) * abs(y2 - y1)
8     events.append((z1, 1, area, i))
9     events.append((z2, -1, area, i))
10 events.sort()
11 min_used = N + 1
12 now_used = 0
13 now_area = 0
14 for event in events:
15     if event[1] == 1:
16         now_used += 1
17         now_area += event[2]
18         if now_area == total_area and now_used < min_used:
19             min_used = now_used
20     else:
21         now_used -= 1
22         now_area -= event[2]
23
24 if min_used == N + 1:
25     print('NO')
26 else:
27     print('YES')
28     print(min_used)
29     used_blocks = set()
30     for event in events:
31         if event[1] == 1:
32             now_used += 1
33             used_blocks.add(event[3])
34             now_area += event[2]
35             if now_area == total_area and now_used == min_used:
36                 print(*used_blocks)
37                 break
38     else:
```

[Отправить](#)

[Предыдущая](#)