

Прямоугольники в квадрате

В двух отраслях — электронике и телекоммуникациях — прогресс материализуется настолько зримо, что начинаешь воспринимать его как природную силу, управляющую судьбами людей и компаний и преобразующую человеческую жизнь. Мэйнфреймы, мини-компьютеры, микрокомпьютеры, LAN, WAN, Интернет, оптоволокно, встроенные системы, широкополосная передача данных — целые эпохи вычислений и коммуникаций сменяли друг друга на отрезке времени намного короче человеческой жизни — длиной всего-то в 30 лет. Прогресс имеет свои жизненные циклы, и периоды роста в производстве полупроводниковых блоков примерно раз в пять лет чередуются с периодами спада. Но это спад и рост только по количеству, многие эксперты сходятся во мнении, что количественный спад — лишь передышка, в течение которой прогресс накапливает силы для нового качественного рывка. Сегодня основной причиной спада в мировой полупроводниковой отрасли эксперты называют недостаток инструментальных средств нового поколения для автоматизированного проектирования электронных систем (EDA, Electronic Design Automation), которые позволили бы учитывать возможности, открывающиеся благодаря последним достижениям в технологиях производства полупроводниковых структур (переход к нанометровым технологиям).

ТЕХНОЛОГИЯ, мкм	КОЛИЧЕСТВО УПАКОВАННЫХ ТРАНЗИСТОРОВ, млн.	ДЛЯ РАЗРАБОТКИ ТРЕБУЕТСЯ	
		человеко-лет	млн. долл.
0,35	2	<1	<1
0,25	8	5	2
0,18	40	50	10
0,15	60	85	20
0,13	>100	>200	>50

На очень общем уровне процесс проектирования электронных систем включает в себя четыре стадии:

- 1) проработка идеи и постановка технического задания;
- 2) создание функциональной схемы;
- 3) разработка принципиальной схемы, размещение компонентов и трассировка проводников на плате;
- 4) физическое воплощение.

В данной задаче нам требуется помочь разработчикам современных EDA с третьим этапом проектирования. Представим подложку, на которой необходимо разместить полупроводниковые элементы, в виде квадрата. У нас также имеется некоторый набор прямоугольных полупроводниковых элементов, которые необходимо разместить на подложке оптимальным образом. При этом элементы не должны пересекаться между собой и выходить за пределы квадрата. Естественно, что место не должно пропадать даром, поэтому, чем меньше площади останется незаполненной, тем компактнее получится разработанная система. Процесс проектирования окажется более эффективным и позволит избежать ненужных затрат на производство лишних подложек!

Входные данные

t — число тестовых последовательностей, затем следуют t тестовых последовательностей. [$t \leq 500$]

На первой строчке каждой тестовой последовательности стоит целое число N , а на второй строчке целое число K . Где N - это длина стороны квадрата [$2 \leq N \leq 1000$], а K - число различных доступных прямоугольников [$1 \leq K \leq 10000$]. Затем следуют ровно K строк, на каждой из которых стоят 3 числа: w_i , h_i , l_i . w_i - длина прямоугольника [$w_i \leq N$], h_i - высота прямоугольника [$h_i \leq N$], l_i - количество таких прямоугольников [$l_i \leq 200000$]. Вы можете вращать прямоугольники на угол кратный 90 градусам.

Выходные данные

Для каждого теста выведите число R - количество использованных прямоугольников, за которым следуют ровно R строк. В каждой строке выведите целые координаты противоположных углов каждого из прямоугольников x_{i1} , y_{i1} , x_{i2} , y_{i2} . Решение будет считаться верным (accepted) если все выведенные прямоугольники не будут пересекаться между собой, и не будут выходить за границы квадрата.

Начисление очков

Общие очки равны сумме очков полученных индивидуально за каждый из тестов. Количество очков за каждый тест равно суммарной покрытой площади квадрата деленное на площадь квадрата. За тест, в котором квадрат с длиной стороны N покрыт целиком, вы получите 4 очка. Если полученные вами очки выглядят как $xxx.xxxxaaa$, то число aaa = числу полностью покрытых квадратов.

Пример

Входные данные:

```
1
10
8
3 5 2
2 2 1
2 3 1
2 5 1
4 5 1
1 3 2
3 8 1
1 1 1
```

Выходные данные:

```
9
1 1 5 3
6 1 8 5
9 1 10 2
1 4 5 7
6 6 10 7
9 3 10 5
1 8 1 10
2 8 2 10
3 8 10 10
```

Пояснение, к примеру:

