

Задача триангуляции

Условия:

- Напишите функцию `triangulation`, которая будет отрисовывать станции и примерное местонахождение пользователя.
- Территорию представляет блок `.Map` размером 500x500px;
- Базовая станция отображается черной (`#000`) точкой 20x20px;
- Радиус покрытия – это окружность толщины 1px и цветом `#000`;
- На вход получаем два списка, первый - список базовых станций, второй - список `id` станций, зарегистрировавших пользователя в определенные моменты (единовременно пользователя регистрируют как минимум две станции);
- Территорию, где находился пользователь в зависимости от регистрации у тех или иных вышек заливаем цветом. Цвет меняется инкрементально, в зависимости от положения элемента во входном списке (прим: 0-е пересечение - `#000`, 3-е - `#333`, 10-е - `#aaa`).

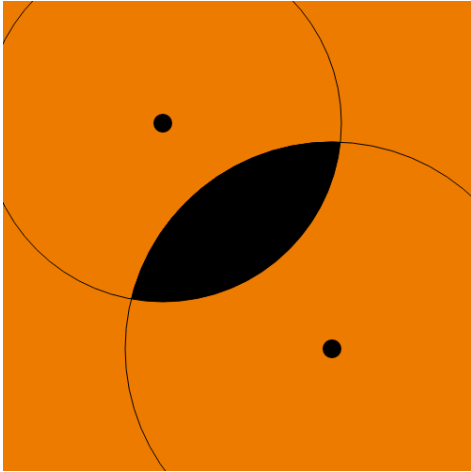
```
interface Station {  
  id: number;  
  x: number;  
  y: number;  
  r: number;  
}
```

Примеры

Скриншоты ожидаемого результата

Пример 1

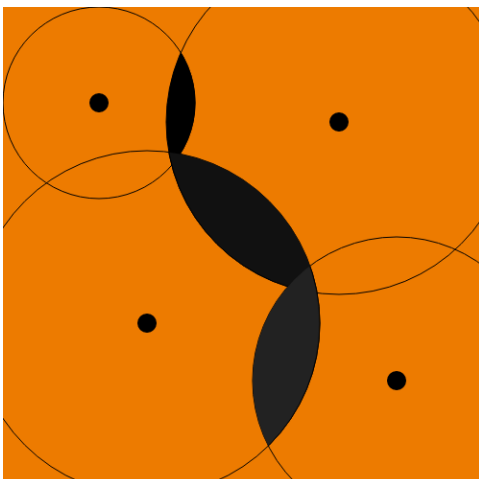
```
const stations = [{  
  "id": 1,  
  "x": 350,  
  "y": 370,  
  "r": 220  
},  
{  
  "id": 2,  
  "x": 170,  
  "y": 130,  
  "r": 190  
}];  
  
const stationIds = [[1, 2]]
```



Пример 2

```
const stations = [{
  "id": 1,
  "x": 100,
  "y": 100,
  "r": 100
}, {
  "id": 2,
  "x": 350,
  "y": 120,
  "r": 180
}, {
  "id": 3,
  "x": 150,
  "y": 330,
  "r": 180
}, {
  "id": 4,
  "x": 410,
  "y": 390,
  "r": 150
}
];
```

```
const stationIds = [
  [1, 2],
  [2, 3],
  [3, 4]
]
```



Пример 3

```
const stations = [{  
  "id": 1,  
  "x": 250,  
  "y": 100,  
  "r": 160  
}, {  
  "id": 2,  
  "x": 125,  
  "y": 325,  
  "r": 160  
}, {  
  "id": 3,  
  "x": 375,  
  "y": 325,  
  "r": 160  
}];  
  
const stationIds = [[1, 2, 3]];
```

