

Чемпионат по программированию: Бэкенд-разработка - Квалификация

🕒 20 окт 2019, 18:01:25

старт: 20 окт 2019, 12:36:19

финиш: 20 окт 2019, 17:36:19

длительность: 05:00:00

начало: 14 окт 2019, 12:00:00

конец: 20 окт 2019, 23:59:59

А. Погоня за высоким спросом

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Есть город, в котором работает сервис такси. Город представлен как матрица длиной N и шириной M . В городе бывают зоны повышенного спроса, т.е. зоны где есть много пассажиров и мало водителей. Коэффициент повышенного спроса для такой зоны считается по формуле $S = K / D$, где K - количество людей, которые хотят заказать машину, и D - количество заехавших в зону водителей.

Нужно найти количество минут, через которое водители погасят коэффициент (т.е. сделают его $= 1$), если все поедут в зону повышенного спроса. За одну минуту водитель проезжает одну ячейку матрицы. Двигаться можно во все стороны, кроме диагональных.

Формат ввода

В первой строчке через пробел задается размерность матрицы города $N * M$, где $N \leq 10000$, $M \leq 10000$

Во второй строчке через пробел представлена геометрия зоны повышенного спроса $x_0 y_0 x_1 y_1$, где x_0 принадлежит $[0, N]$, x_1 принадлежит $[0, N+1]$, y_0 принадлежат $[0, M]$, а y_1 принадлежат $[0, M+1]$, притом прямоугольник не включает $x_1 y_1$ и $x_0 < x_1$, $y_0 < y_1$.

В третьей строчке через пробел задается число пользователей в зоне повышенного спроса K и число водителей D , которые могут поехать в зоны повышенного спроса. ($K \leq 300000$, $D \leq 300000$)

В четвертой строчке через пробел задаются координаты водителей вне зоны спроса: $x_0 y_0 x_1 y_1 \dots$

Формат вывода

Выведите минимальное количество минут, которые потребуются водителям, чтобы снять зону повышенного спроса (т.е. сделать $S = 1$). Если решения не существует, нужно вывести "-1".

Пример 1

Ввод	Вывод
2 2 1 1 2 2 1 1 0 0	2

Пример 2

Ввод	Вывод
2 1 1 0 2 1 1 1 0 0	1

Язык Python 3.7.3

Набрать здесь Отправить файл

```
1 def delta_s(coord_D: int, coord_0: int, coord_1: int) -> int:
2     delta = 0
3     if coord_D < coord_0:
4         delta = coord_0 - coord_D
5     elif coord_0 <= coord_D < coord_1:
6         delta = 0
7     elif coord_1 <= coord_D:
8         delta = coord_D - (coord_1 - 1)
9     return delta
10
11
12 def main():
13     with open('input.txt', 'r') as file:
14         info = file.readlines()
15
16     K, D = list(map(lambda x: int(x), info[2].split()))
17     available_cars = list(map(lambda x: int(x), info[3].split()))
18     if K <= D:
19         demand_zone = info[1].split()
20
21         x0, y0, x1, y1 = map(lambda x: int(x), demand_zone)
22         available_cars_coords = list(zip(available_cars[:2], available_cars[1::2]))
23         cars_times = []
24         for i in available_cars_coords:
25             time = delta_s(i[0], x0, x1) + delta_s(i[1], y0, y1)
26             cars_times.append(time)
27
28         cars_times.sort()
29         return max(cars_times[:K])
30     else:
31         return -1
32
33
34 if __name__ == '__main__':
35     print(main())
36
```

Отправить

Следующая