

# Тренировки по алгоритмам 4.0 от Яндекса — Занятие 3 (Кратчайшие пути в графах)

2 апр 2024, 06:39:04  
старт: 8 ноя 2023, 22:00:00  
...

Объявления жюри

Положение участников ЗадачиПосылки

## D. Автобусы в Васюках

|                     |                                  |            |
|---------------------|----------------------------------|------------|
|                     | Все языки                        | Python 3.6 |
| Ограничение времени | 3 секунды                        | 5 секунд   |
| Ограничение памяти  | 512Mb                            | 256Mb      |
| Ввод                | стандартный ввод или input.txt   |            |
| Вывод               | стандартный вывод или output.txt |            |

- ✓

[A. Дейкстра](#)
- ✓

[B. Дейкстра с восстановлением пути](#)
- ✓

[C. Быстрый алгоритм Дейкстры](#)
- ✓

[D. Автобусы в Васюках](#)
- ✓

[E. На санях](#)

Между некоторыми деревнями края Васюки ходят автобусы. Поскольку пассажиропотоки здесь не очень большие, то автобусы ходят всего несколько раз в день. Марии Ивановне требуется добраться из деревни d в деревню v как можно быстрее (считается, что в момент времени 0 она находится в деревне d).

### Формат ввода

Сначала вводится число N – общее число деревень ( $1 \leq N \leq 100$ ), затем номера деревень d и v, за ними следует количество автобусных рейсов R ( $0 \leq R \leq 10000$ ). Далее идут описания автобусных рейсов. Каждый рейс задается номером деревни отправления, временем отправления, деревней назначения и временем прибытия (все времена – целые от 0 до 10000). Если в момент t пассажир приезжает в какую-то деревню, то уехать из нее он может в любой момент времени, начиная с t.

### Формат вывода

Выведите минимальное время, когда Мария Ивановна может оказаться в деревне v. Если она не сможет с помощью указанных автобусных рейсов добраться из d в v, выведите -1.

### Пример

| Ввод   | Вывод |
|--|-------|
| 3<br>1 3<br>4<br>1 0 2 5<br>1 1 2 3<br>2 3 3 5<br>1 1 3 10 | 5     |

Язык Python 3.12.1

Набрать здесь Отправить файл

```
1 import heapq
2 from collections import defaultdict
3
4 def getNextPoints(graph, curr_time, curr_point):
5     res = []
6     for end_time, start_time, next_point in graph[curr_point]:
7         if curr_time <= start_time: res.append((end_time, next_point))
8
9     return res
10
11 fin = open('input.txt')
12 N = int(fin.readline())
13 start, end = [int(x) for x in fin.readline().split()]
14
15 R = int(fin.readline())
16 graph = defaultdict(list)
17 for _ in range(R):
18     x, t0, y, t1 = [int(x) for x in fin.readline().split()]
19     graph[x].append((t1, t0, y))
20
21 visited = set()
22 time = defaultdict(lambda: float('inf'))
23 time[start] = 0
24
25 stack = [(0, start)]
26 while stack:
27     curr_time, curr_point = heapq.heappop(stack)
28
29     if curr_point not in visited:
30         for end_time, next_point in getNextPoints(graph, curr_time, curr_point):
31             if end_time < time[next_point]:
32                 time[next_point] = end_time
33                 heapq.heappush(stack, (end_time, next_point))
34
35     visited.add(curr_point)
36
37 print(time[end]) if time[end] != float('inf') else print(-1)
```

Отправить 1 осталось 98 попыток

Предыдущая

Следующая

| Время посылки         | ID       | Задача | Компилятор               | Вердикт | Тип посылки | Время | Память  | Тест | Баллы |
|-----------------------|----------|--------|--------------------------|---------|-------------|-------|---------|------|-------|
| 11 ноя 2023, 04:26:11 | 96776527 | D      | Python 3.12.1            | OK      | -           | 73ms  | 4.30Mb  | -    | -     |
| 11 ноя 2023, 03:04:57 | 96774306 | D      | Python 3.9 (PyPy 7.3.11) | OK      | -           | 264ms | 28.09Mb | -    | -     |