

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

2 апр 2024, 06:38:48  
старт: 8 ноя 2023, 22:00:00  
...

Объявления жюри

Положение участников    Задачи    Посылки

## В. Дейкстра с восстановлением пути

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

- ✓

[A. Дейкстра](#)
- ✓

**В. Дейкстра с восстановлением пути**
- ✓

[C. Быстрый алгоритм Дейкстры](#)
- ✓

[D. Автобусы в Васюках](#)
- ✓

[E. На санях](#)

Дан ориентированный взвешенный граф. Найдите кратчайший путь от одной заданной вершины до другой.

### Формат ввода

В первой строке содержатся три числа: N, S и F ( $1 \leq N \leq 100$ ,  $1 \leq S, F \leq N$ ), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках вводится по N чисел, не превосходящих 100, — матрица смежности графа, где -1 означает, что ребра между вершинами нет, а любое неотрицательное число — наличие ребра данного веса. На главной диагонали матрицы записаны нули.

### Формат вывода

Последовательно выведите все вершины одного (любого) из кратчайших путей, или -1, если пути между указанными вершинами не существует

### Примечания

Пример ввода:

3 2 1  
  
0 1 1  
  
4 0 1  
  
2 1 0

Пример вывода:

2 3 1

Язык Python 3.12.1

Набрать здесь    Отправить файл

```
1 import heapq
2 from collections import deque
3
4 def getNextPoints(graph, curr_point):
5     res = []
6
7     for i, dist in enumerate(graph[curr_point]):
8         if (i != curr_point) and (dist >= 0): res.append((dist, i))
9
10    return res
11
12 fin = open('input.txt')
13 N, start, end = [int(x) if i == 0 else int(x)-1 for i, x in enumerate(fin.readline().split())]
14 graph = [[int(x) for x in fin.readline().split()] for _ in range(N)]
15
16 visited = set()
17 distance = [float('inf')]*N
18 distance[start] = 0
19 prev_points = [-1]*N
20 prev_points[start] = start
21
22 stack = [(0, start)]
23 while stack:
24     curr_dist, curr_point = heapq.heappop(stack)
25
26     if curr_point not in visited:
27         for next_dist, next_point in getNextPoints(graph, curr_point):
28             if curr_dist+next_dist < distance[next_point]:
29                 distance[next_point] = curr_dist+next_dist
30                 prev_points[next_point] = curr_point
31                 heapq.heappush(stack, (curr_dist+next_dist, next_point))
32
33     visited.add(curr_point)
34
35 if prev_points[end] != -1:
36     path, curr_point = deque(), end
37     while curr_point != start:
38         path.appendleft(curr_point)
39         curr_point = prev_points[curr_point]
39     path.appendleft(start)
```

Отправить 

осталось 98 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
11 ноя 2023, 04:26:41	96776537	B	Python 3.12.1	OK	-	58ms	4.14Mb	-	- <a href="#">отчёт</a>
11 ноя 2023, 02:15:08	96771926	B	Python 3.9 (PyPy 7.3.11)	OK	-	242ms	28.09Mb	-	- <a href="#">отчёт</a>