

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

⌚ 4 апр 2024, 02:44:21


старт: 28 окт 2023, 21:14:17

финиш: 26 ноя 2023, 03:14:16

длительность: 28д. 5ч.

...

Объявления жюри

 Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

[Положение участников](#) [Задачи](#) [Посылки](#)

37. Путь в графе

	Все языки	Python 3.6
Ограничение времени	1 секунда	5 секунд
Ограничение памяти	64Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

В неориентированном графе требуется найти минимальный путь между двумя вершинами.



Формат ввода

Первым на вход поступает число N — количество вершин в графе ($1 \leq N \leq 100$). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 — наличие ребра). Далее задаются номера двух вершин — начальной и конечной.

Формат вывода

Выведите сначала L — длину кратчайшего пути (количество ребер, которые нужно пройти), а потом сам путь. Если путь имеет длину 0, то его выводить не нужно, достаточно вывести длину.
Необходимо вывести путь (номера всех вершин в правильном порядке). Если пути нет, нужно вывести -1.

Пример

Ввод 	Вывод 
10 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 5 4	2 5 2 4

Язык

Python 3.12.1

Набрать здесь

Отправить файл

```
1 from collections import deque
2
3 fin = open('input.txt')
4 N = int(fin.readline())
5
6 array = [0]*N
7 for i in range(N):
8     array[i] = [int(x) for x in fin.readline().split()]
9
10 start, end = [int(x)-1 for x in fin.readline().split()]
11
12 def getShortPath(array, start, end):
13     if start == end: return 0
14
15     path = dict()
16     stack, visited = deque([start]), set([start])
17     while stack:
18         curr_pos = stack.pop()
19
20         for next_pos in range(len(array)):
21             if (array[curr_pos][next_pos] == 1) and (next_pos not in visited):
22                 if next_pos == end: return path, curr_pos, next_pos
23                 stack.appendleft(next_pos)
24                 visited.add(next_pos)
25                 path[next_pos] = curr_pos
26
27     return -1
28
29 path = getShortPath(array, start, end)
30 if path == -1:
31     print(-1)
32 elif path == 0:
33     print(0)
34 else:
35     path, prev, last = path
36
37     ans = [last]
38     while prev != start:
```

Отправить

 осталось 100 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
30 окт 2023, 01:50:12	95152534	37	Python 3.12.1	OK	-	56ms	4.26Mb	-	-	отчёт
30 окт 2023, 01:48:37	95152385	37	Python 3.12.1	RE	-	59ms	4.14Mb	18	-	отчёт
30 окт 2023, 01:48:06	95152338	37	Python 3.12.1	PE	-	52ms	4.26Mb	1	-	отчёт

-  1. Гистограмма
-  2. Красивая строка
-  3. Коллекционер Диего
-  4. Контрольная работа
-  5. Хорошая строка
-  6. Операционные системы lite
-  7. SNTP
-  8. Минимальный прямоугольник
-  9. Сумма в прямоугольнике
-  10. Скучная лекция
-  11. Стек с защитой от ошибок
-  12. Правильная скобочная последовательность
-  13. Постфиксная запись
-  14. Сортировка вагонов lite
-  15. Великое Лайнландское переселение
-  16. Очередь с защитой от ошибок
-  17. Игра в пьяницу
-  18. Дек с защитой от ошибок
-  19. Хипуй
-  20. Пирамидальная сортировка
-  21. Три единицы подряд
- [22. Кузнечик](#)
- [23. Калькулятор](#)
- [24. Покупка билетов](#)
- [25. Гвоздики](#)
- [26. Самый дешёвый путь](#)
- [27. Вывести маршрут максимальной стоимости](#)
- [28. Ход конём](#)
- [29. Кафе](#)
- [30. НОП с восстановлением ответа](#)
-  31. Поиск в глубину
-  32. Компоненты связности
-  33. Списывание
-  34. Топологическая сортировка
-  35. Поиск цикла
-  36. Длина кратчайшего пути
-  37. Путь в графе
-  38. Блохи
-  39. Путь спелеолога
-  40. Метро