

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

4 апр 2024, 02:44:02
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 участниковЗадачиПосылки

35. Поиск цикла

	Все языки	Python 3.6
Ограничение времени	2 секунды	5 секунд
Ограничение памяти	256Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Дан неориентированный граф. Требуется определить, есть ли в нем цикл, и, если есть, вывести его.

Формат ввода

В первой строке дано одно число n — количество вершин в графе ($1 \leq n \leq 500$). Далее в n строках задан сам граф матрицей смежности.

Формат вывода

Если в исходном графе нет цикла, то выведите «NO». Иначе, в первой строке выведите «YES», во второй строке выведите число k — количество вершин в цикле, а в третьей строке выведите k различных чисел — номера вершин, которые принадлежат циклу в порядке обхода (обход можно начинать с любой вершины цикла). Если циклов несколько, то выведите любой.

Пример 1

Ввод	Вывод
3 0 1 1 1 0 1 1 1 0	YES 3 3 2 1

Пример 2

Ввод	Вывод
4 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0	NO

Пример 3

Ввод	Вывод
5 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 1 0	YES 3 5 4 3

ЯзыкPython 3.9 (PyPy 7.3.11)

Набрать здесьОтправить файл

```
1 fin = open('input.txt')
2 N = int(fin.readline())
3
4 matrix = [[] for _ in range(N)]
5 for i, line in enumerate(fin):
6     matrix[i] = [int(x) for x in line.split()]
7
8 def answer(matrix):
9     if len(matrix) == 1: return False
10    visited, path = set(), dict()
11
12    for i in range(N):
13        if i not in visited:
14            visited.add(i)
15
16            stack = [(i, None)]
17            while stack:
18                curr_v, prev_v = stack.pop()
19
20                for next_v in range(N):
21                    if matrix[curr_v][next_v] == 1:
22                        if next_v not in visited:
23                            visited.add(next_v)
24                            stack.append((next_v, curr_v))
25                            path[next_v] = curr_v
26                        elif prev_v != next_v:
27                            return (path, curr_v, next_v)
28
29    return False
30
31 res = answer(matrix)
32 if res:
33     path, prev_v, last_v = res
34
35     ans = [path[last_v], last_v]
36     while prev_v != ans[0]:
37         ans.append(prev_v)
38         prev_v = path[prev_v]
```

Отправить1 осталось 100 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
30 окт 2023, 01:05:13	95147941	35	Python 3.9 (PyPy 7.3.11)	OK	-	226ms	28.09Mb	-	-	отчёт
30 окт 2023, 00:34:41	95144000	35	Python 3.12.1	OK	-	106ms	4.86Mb	-	-	отчёт
29 окт 2023, 23:32:16	95135515	35	Python 3.12.1	WA	-	52ms	4.20Mb	10	-	отчёт
29 окт 2023, 23:18:32	95133599	35	Python 3.12.1	WA	-	56ms	4.15Mb	10	-	отчёт
29 окт 2023, 23:17:32	95133440	35	Python 3.12.1	RE	-	48ms	4.25Mb	8	-	отчёт
29 окт 2023, 23:15:39	95133190	35	Python 3.12.1	RE	-	48ms	4.25Mb	7	-	отчёт
29 окт 2023, 23:15:04	95133101	35	Python 3.12.1	WA	-	50ms	3.93Mb	1	-	отчёт
29 окт 2023, 22:39:35	95128137	35	Python 3.12.1	RE	-	53ms	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