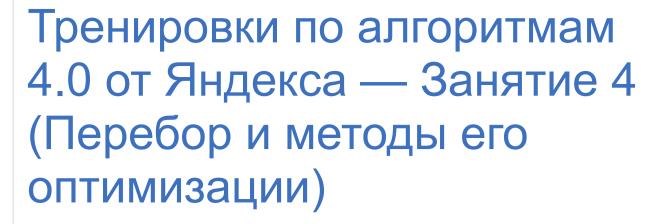
Пробный контест Архив соревнований Настройки компиляторов Значения ошибок Команды



② 2 апр 2024, 06:42:44 Объявления жюри старт: 18 ноя 2023, 15:00:00

А. Все перестановки

заданной длины

В. Затерянный мир

D. Простая задача

коммивояжера

Е. Генерация

С. Максимальный разрез

правильных скобочных

последовательностей - 2

Положение участников Задачи Посылки

D. Простая задача коммивояжера

,	Ограничение времени	2 секунды
	Ограничение памяти	64Mb
	Ввод	стандартный ввод или input.txt
	Вывод	стандартный вывод или output.txt

Неориентированный взвешенный граф задан матрицей смежности. Найдите кратчайший цикл, который начинается и заканчивается в вершине номер 1 и проходит через все вершины по одному разу.

Формат ввода

В первой строке вводится число N ($N \le I0$) — количество вершин графа. Следующие N строк содержат по N целых неотрицательных чисел и задают матрицу смежности. Число 0 означает, что ребро отстутствует. Любое другое число задаёт вес ребра.

Формат вывода

Выведите минимальную суммарную длину цикла или число -1, если цикл построить невозможно.

Пример 1

 Вывод
 П

 1
 0

 0
 0

Пример 2

 Ввод П

 2
 2

 0 1
 1 0

Пример 3

 Ввод П

 2

 0 85

 85 0

Язык Python 3.9 (PyPy 7.3.11) ~

Набрать здесь

Отправить

Отправить файл

🗓 осталось 98 попыток

```
1 fin = open('input.txt')
2 N = int(fin.readline())
 graph = [None]*N
for i in range(N):
    graph[i] = [int(x) for x in fin.readline().split()]
 def findMinPath(graph, start_point, end_point):
    def findPath(curr_point, curr_dist, visited_count, all_points, remaining_path):
10
                nonlocal best_path
11
              if visited_count == all_points:
    if graph[curr_point][end_point] != 0:
        best_path = min(best_path, curr_dist+graph[curr_point][end_point])
        return curr_dist+graph[curr_point][end_point]
12
13
14
15
16
17
                           return float('inf')
18
                else:
19
                     min_path = float('inf')
20
21
22
                     for i in range(len(graph[0])):
   if not visited[i] and graph[curr_point][i] != 0:
23
                                 visited[i] = True
                                 if curr_dist+remaining_path < best_path:
    min_path = min(min_path, findPath(i, curr_dist+graph[curr_point][i</pre>
24
25
26
                                 visited[i] = False
27
28
                     return min_path
29
30
          N = len(graph[0])
31
32
33
34
35
          if N == 1: return 0
          if N == 2:
               if graph[0][1] != 0 and graph[1][0] != 0:
                     return graph[0][1]+graph[1][0]
                else:
36
                     return -1
37
38 <
                                                                                                                                  >
```

Предыдущая Следующая Время посылки Задача Компилятор Вердикт Тип посылки Баллы 22 ноя 2023, 04:05:20 98417395 D Python 3.9 (PyPy 7.3.11) OK 28.10Mb отчёт 98416196 OK 28.09Mb 22 ноя 2023, 03:29:37 Python 3.9 (PyPy 7.3.11) 404ms отчёт