

Тренировки по алгоритмам 4.0 от Яндекса — Занятие 4 (Перебор и методы его оптимизации)

2 апр 2024, 06:42:35
старт: 18 ноя 2023, 15:00:00
...

Объявления жюри

Положение участников Задачи Посылки

С. Максимальный разрез

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

Взвешенный неориентированный граф без петель задан матрицей смежности. Распределите вершины по двум долям так, чтобы сумма весов рёбер, соединяющих вершины из разных долей, была максимальна.

Формат ввода

Вводится число N ($2 \leq N \leq 20$) — количество вершин в графе.
В следующих N строках, содержащих по N целых чисел от 0 до 1000, задаётся матрица смежности. 0 означает отсутствие ребра.

Формат вывода

В первой строке выведите суммарный вес рёбер, соединяющих вершины из разных долей.
Во второй строке выведите N чисел 1 или 2 — номера долей для каждой из вершин графа.

Пример 1

Ввод	Вывод
2 0 1 1 0	1 2 1

Пример 2

Ввод	Вывод
3 0 1 2 1 0 2 2 2 0	4 2 2 1

Пример 3

Ввод	Вывод
4 0 10 3 0 10 0 7 2 3 7 0 9 0 2 9 0	26 2 1 2 1

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

```
1 def partition(curr_point, curr_sum, v_number):
2     if curr_point == v_number:
3         return curr_sum, teams.copy()
4     else:
5         teams[curr_point] = 1
6         temp_sum = curr_sum
7         for i in range(v_number):
8             if graph[curr_point][i] != 0 and teams[i] == 2:
9                 temp_sum += graph[curr_point][i]
10                max_sum1, parts1 = partition(curr_point+1, temp_sum, v_number)
11
12                teams[curr_point] = 2
13                temp_sum = curr_sum
14                for i in range(v_number):
15                    if graph[curr_point][i] != 0 and teams[i] == 1:
16                        temp_sum += graph[curr_point][i]
17                max_sum2, parts2 = partition(curr_point+1, temp_sum, v_number)
18
19                teams[curr_point] = 0
20
21                if max_sum1 > max_sum2:
22                    return max_sum1, parts1
23            else:
24                return max_sum2, parts2
25
26 fin = open('input.txt')
27 N = int(fin.readline())
28 graph = [None]*N
29
30 for i in range(N):
31     graph[i] = [int(x) for x in fin.readline().split()]
32
33 teams = [0]*N
34 teams[0] = 1
35
36 # Real Max
37 max_sum, parts = partition(1, 0, N)
38 print(max_sum)
```

Отправить

осталось 95 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
23 ноя 2023, 01:12:59	98641995	C	Python 3.9 (PyPy 7.3.11)	OK	-	0.549s	28.09Mb	-	-
23 ноя 2023, 01:10:15	98641648	C	Python 3.9 (PyPy 7.3.11)	OK	-	0.561s	28.10Mb	-	-
23 ноя 2023, 01:09:54	98641601	C	Python 3.9 (PyPy 7.3.11)	PE	-	187ms	28.09Mb	1	-
22 ноя 2023, 05:46:55	98418943	C	Python 3.12.1	TL	-	1.088s	4.26Mb	15	-
22 ноя 2023, 05:46:21	98418940	C	Python 3.9 (PyPy 7.3.11)	OK	-	0.906s	28.48Mb	-	-