Graph [14]

A. Алгоритм Форда — Беллмана

1 секунда, 256 мегабайт

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти длины кратчайших путей от заданной вершины до всех остальных.

Входные данные

В первой строке вам дано числа n, m и s $(1 \le n, m \le 10^4, 1 \le s \le n)$ — количество вершин в графе, количество рёбер в графе, стартовая вершина. В следующих m строках вам даны рёбра в виде троек чисел u, v, w $(1 \le u, v \le n, 0 \le w \le 10^5)$ — пара вершин соединяемых ребром и его длина.

Выходные данные

Выведите n чисел d_i — длины кратчайших путей из заданной стартовой вершины до вершины i.

входные данные

```
5 4 1
1 4 3
1 3 1
3 4 1
5 4 2
```

выходные данные

```
0 -1 1 2 4
```

B. Алгоритм Дейкстры

1 секунда, 256 мегабайт

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти длины кратчайших путей от заданной вершины до всех остальных.

Входные данные

В первой строке вам дано числа n, m и s $(1 \le n \le 10^4, 1 \le m \le 5 \cdot 10^5, 1 \le s \le n)$ — количество вершин в графе, количество рёбер в графе, стартовая вершина. В следующих m строках вам даны рёбра в виде троек чисел u, v, w $(1 \le u, v \le n, 0 \le w \le 10^5)$ — пара вершин соединяемых ребром и его длина.

Выходные данные

Выведите n чисел d_i — длины кратчайших путей из заданной стартовой вершины до вершины i.

входные данные

```
5 4 1
1 4 3
1 3 1
3 4 1
5 4 2
```

выходные данные

```
0 -1 1 2 4
```

C. Алгоритм Дейкстры!

1 секунда, 256 мегабайт

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти длины кратчайших путей от заданной вершины до всех остальных.

Входные данные

В первой строке вам дано числа n, m и s $(1 \le n \le 10^5, 1 \le m \le 5 \cdot 10^5, 1 \le s \le n)$ — количество вершин в графе, количество рёбер в графе, стартовая вершина. В следующих m строках вам даны рёбра в виде троек чисел u, v, w $(1 \le u, v \le n, 0 \le w \le 10^5)$ — пара вершин соединяемых ребром и его длина.

Выходные данные

Выведите n чисел d_i — длины кратчайших путей из заданной стартовой вершины до вершины i.

входные данные

```
5 4 1
1 4 3
1 3 1
3 4 1
5 4 2
```

выходные данные

```
0 -1 1 2 4
```

D. Алгоритм Дейкстры?

1 second, 64 megabytes

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти кратчайший путь из вершины 1 в вершину n.

Входные данные

В первой строке содержатся целые числа n и m $(2 \le n \le 10^5, 0 \le m \le 10^5)$, где n — количество вершин, а m — количество ребер в графе. Далее в m строках содержатся сами ребра, по одному в строке. Каждое ребро задается тремя числами a_i, b_i, w_i $(1 \le a_i, b_i \le n, 1 \le w_i \le 10^6)$, где a_i, b_i — это концы ребра, а w_i — его длина.

Граф может содержать кратные ребра и петли.

Выходные данные

Выведите число `-1` если пути нет или сам кратчайший путь, если он существует.

входные данные

```
5 6
1 2 2
2 5 5
2 3 4
1 4 1
4 3 3
3 5 1
```

выходные данные

```
1 4 3 5
```

входные данные

```
5 6
1 2 2
2 5 5
2 3 4
1 4 1
4 3 3
3 5 1
```

выходные данные

```
1 4 3 5
```

E. Алгоритмы Дейкстры

3 секунды, 256 мегабайт

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти длины кратчайших путей от заданных вершин до всех остальных.

Входные данные

В первой строке вам даны два числа n и q $(1 \le n \le 2500,\ 1 \le q \le 20)$ — количество вершин в графе и количество запросов. В следующих n строках вам дана матрица расстояний между вершинами a_{ij} $(0 \le a_{ij} \le 10^5)$. В следующих q строках даны числа s_i $(1 \le s_i \le n)$ — стартовые вершины.

Выходные данные

Для каждого запроса выведите на отдельной строке n чисел d_i — длины кратчайших путей из заданной стартовой вершины до вершины i.

входные данные

```
3 3
0 1 3
1 0 1
3 1 0
1
2
3
```

выходные данные

```
0 1 2
1 0 1
2 1 0
```

F. Количество двоичных деревьев

2 секунды, 64 мегабайта

Посчитайте количество двоичных деревьев с заданным количеством вершин, так как их количество может быть слишком велико, выведите результат по заданному модулю.

Входные данные

Вам даны два числа n и m $(0 \le n \le 10^4,\ 1 \le m \le 10^6 + 3)$.

Выходные данные

Выведите количество двоичных деревьев с n вершинами по модулю m.

входные данные

```
1 1000003
```

выходные данные

```
1
```

входные данные

```
2 1000003
```

выходные данные

```
2
```

входные данные

```
3 1000003
```

выходные данные

```
5
```

входные данные

```
4 1000003
```

выходные данные

```
14
```

входные данные

```
5 1000003
```

выходные данные

```
42
```

G. Широчайший путь

1 секунда, 256 мегабайт

Вам дан неориентированный взвешенный граф. Веса рёбер обозначают их ширину. Шириной пути называется ширина наименьшего ребра на пути. Для каждой вершины найдите максимальную ширину пути от стартовой вершины до неё.

Входные данные

В первой строке вам дано числа n, m и s $(1 \le n \le 10^5,\ 1 \le m \le 5 \cdot 10^5,\ 1 \le s \le n)$ — количество вершин в графе, количество рёбер в графе, стартовая вершина. В следующих m строках вам даны рёбра в виде троек чисел u, v, w $(1 \le u, v \le n,\ 1 \le w \le 10^8)$ — пара вершин соединяемых ребром и его ширина.

Выходные данные

Выведите n чисел d_i — ширину наиболее широкого пути из заданной стартовой вершины до вершины i. Для самой стартовой вершины выведите "-1".

входные данные

```
6 5 3
1 2 10
2 3 20
3 4 30
4 5 10
5 1 20
```

выходные данные

```
10 20 -1 30 10 0
```

H. Алгоритм Флойда — Уоршелла

1 секунда, 256 мегабайт

Задан неориентированный взвешенный граф, вершины которого пронумерованы от 1 до n. Ваша задача найти длины кратчайших путей между всеми парами вершин.

Входные данные

В первой строке вам дано число n $(1 \le n \le 500)$ — количество вершин в графе. В следующих n строках вам даны по n чисел a_{ij} $(0 \le a_{ij} \le 10^9,\ a_{ii} = 0)$ — длины рёбер из i-ой вершины в j-ую.

Выходные данные

Выведите n строк по n чисел d_{ij} — длины кратчайших путей из вершины i в вершину j.

входные данные

```
1
0
```

выходные данные

```
0
```

входные данные

```
3
0 10 1
1 0 1
1 1 0
```

выходные данные

```
0 2 1
1 0 1
1 1 0
```