

36. Длина кратчайшего пути

	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 длину минимального пути 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 L – длину кратчайшего пути (количество ребер, которые нужно пройти). Если пути нет, нужно вывести -1.

Пример 1

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 0 1 0 0 0 1 0
0 1 0 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

Пример 2

Ввод

Вывод

```
5
0 1 0 0 1
1 0 1 0 0
0 1 0 0 0
0 0 0 0 0
1 0 0 0 0
3 5
```

3

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4
```

```
5 int main() {
6     int n;
7     std::cin >> n;
8
9     std::vector<std::vector<int>>> graph(n);
10    for (int i = 0; i < n; i++) {
11        for (int j = 0, tmp; j < n; j++) {
12            std::cin >> tmp;
13            if (tmp == 1) {
14                graph[i].push_back(j);
15            }
16        }
17    }
18
19    int start, end;
20    std::cin >> start >> end;
21    --start, --end;
22
23    if (start == end) {
24        std::cout << 0;
25        return 0;
26    }
27
28    std::vector<int> dist(n, -1);
29
30    std::queue<int> q;
31    q.push(start);
32    dist[start] = 0;
33
34    while (!q.empty()) {
35        int pos = q.front();
36        q.pop();
37
38        for (int curr : graph[pos]) {
```

[Отправить](#)[Предыдущая](#)[Следующая](#)