

Отчёт по расчётной работе по дисциплине ПиОИВИС

Графы

Цель

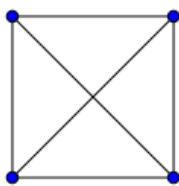
Определить число рёберной связности путём создания алгоритма, который будет создавать граф по заданным параметрам и вычислять сколько нужно удалить рёбер, чтобы он перестал быть связным.

Вариант

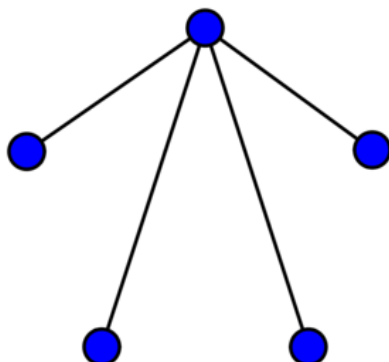
7.2 (мс,нг)

Определения

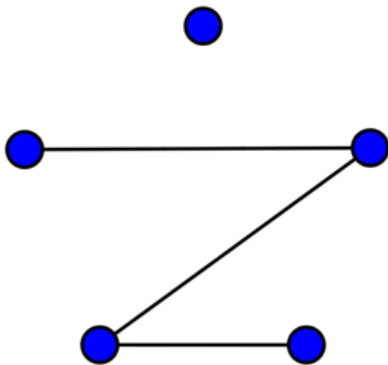
- **Граф** — математическая абстракция реальной системы любой природы, объекты которой обладают парными связями. Граф как математический объект есть совокупность двух множеств — множества самих объектов, называемого множеством вершин, и множества их парных связей, называемого множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.



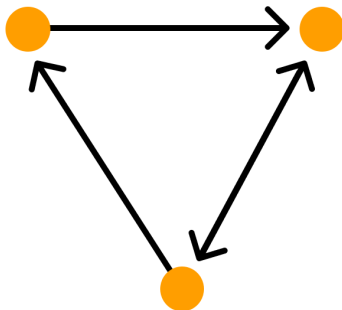
- **Связный граф** — это граф, у которого две любые вершины соединены путём.



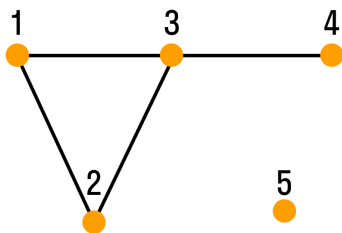
- **Несвязный граф** — граф, содержащий более одной компоненты связности.



- **Ориентированный граф** — это граф, рёбра которого ориентированы, то есть имеют начальную вершину и конечную вершину.

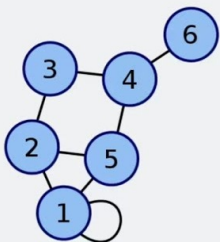


- **Неориентированный граф** — графы, в которых все рёбра являются звеньями, то есть порядок двух концов ребра графа не существует.



Неориентированный граф

- **Смежность** — непосредственная близость, примыкание. В теории графов смежность вершин соответствует наличию ребра между ними.
- **Матрица смежности** - это вид представления графа в виде матрицы, когда пересечение столбцов и строк задаёт дуги.

| Граф | Матрица смежности |
|---|--|
|  | $\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ |

Алгоритм

1. Ввести количество вершин.
2. Ввести количество рёбер.
3. Ввести сами рёбра.
4. Совершить поиск вглубь через dfs и пометить посещённые вершины.
5. Если были посещены все вершины, то граф - связный. В противном случае - несвязный.
6. Пройтись по графу путё удаления и восстановления каждого ребра.
7. Определить минимальное число рёбер, которые нужно удалить что граф из связного превратился в несвязный.
8. Вывести сообщение: "Минимальное количество рёбер, которые нужно удалить для разъединения графа: ".

Код

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  class Graph {
6  public:
7      int V;
8      vector<vector<int>> adj;
9
10     Graph(int V) {
11         this->V = V;
12         adj = vector<vector<int>>(V, vector<int>(V, 0));
13     }
14
15     void addReb(int u, int v) {
16         adj[u][v] = 1;
17         adj[v][u] = 1;
18     }

```

```

19
20 void DFS(int v, vector<bool>& visited) {
21     visited[v] = true;
22     for (int i = 0; i < V; i++) {
23         if (adj[v][i] && !visited[i]) {
24             DFS(i, visited);
25         }
26     }
27 }
28
29 bool isCon() {
30     vector<bool> visited(V, false);
31     DFS(0, visited);
32
33     for (int i = 0; i < V; i++) {
34         if (!visited[i]) return false;
35     }
36     return true;
37 }
38
39 int minRebToRem() {
40     int RebRem = 0;
41
42     for (int u = 0; u < V; u++) {
43         for (int v = u + 1; v < V; v++) {
44             if (adj[u][v]) {
45
46                 adj[u][v] = 0;
47                 adj[v][u] = 0;
48
49                 if (!isCon()) {
50                     RebRem++;
51                 }
52                 adj[u][v] = 1;
53                 adj[v][u] = 1;
54             }
55         }
56     }
57     return RebRem;
58 }
59 };
60
61 int main() {
62     int V, E;
63     cout << "Enter number of vertices: ";
64     cin >> V;
65     Graph g(V);
66
67     cout << "Enter number of edges: ";
68     cin >> E;
69

```

```

70     cout << "Enter edges (u v):" << endl;
71     for (int i = 0; i < E; i++) {
72         int u, v;
73         cin >> u >> v;
74         g.addEdge(u, v);
75     }
76
77     int result = g.minRebToRem();
78     cout << "Minimum number of edges that need to be removed to
79         disconnect a graph:" << result << endl;
80     return 0;
}

```

Примеры работы кода

№1

```

Введите количество вершин: 6
Введите количество рёбер: 9
Введите рёбра (u v):
0 1
0 2
1 3
1 2
1 4
2 4
2 5
3 4
4 5
Минимальное количество рёбер, которые нужно удалить для разъединения графа: 2

```

№2

```

Введите количество вершин: 5
Введите количество рёбер: 6
Введите рёбра (u v):
0 1
1 2
2 0
2 3
4 0
4 3
Минимальное количество рёбер, которые нужно удалить для разъединения графа: 2

```

№3

```

Введите количество вершин: 5
Введите количество рёбер: 7
Введите рёбра (u v):
0 1
0 4
1 2
1 4
4 3
3 2
2 4
Минимальное количество рёбер, которые нужно удалить для разъединения графа: 2

```

№4

```
Введите количество вершин: 4
Введите количество рёбер: 6
Введите рёбра (u v):
0 1
1 2
2 3
3 0
0 2
1 3
Минимальное количество рёбер, которые нужно удалить для разъединения графа: 3
```

№5

```
Введите количество вершин: 8
Введите количество рёбер: 20
Введите рёбра (u v):
0 1
0 7
0 2
0 6
0 4
1 7
1 5
1 3
1 2
2 6
2 3
2 4
3 7
3 5
3 4
4 5
4 6
5 7
5 6
6 7
Минимальное количество рёбер, которые нужно удалить для разъединения графа: 5
```

Вывод

В результате выполнения данной работы были получены следующие практические навыки:

- Изучены основы теории графов.
- Изучены способы представления графов.
- Изучены базовые алгоритмы для работы с графами.

Источники:

<https://skysmart.ru/articles/mathematic/osnovnye-ponyatiya-teorii-grafov>
<https://skyeng.ru/magazine/wiki/it-industriya/cto-takoe-graf/>
<https://habr.com/ru/companies/otus/articles/568026/>
<https://skillbox.ru/media/code/teoriya-grafov-derevya-planarnost-raznovidnosti-grafov/>