РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №4.3

по дисциплине: Дискретная математика
тема: «Графы: Связность»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович

Проверили:
Рязанов Юрий Дмитриевич
Бондаренко Татьяна Владимировна

Белгород 2022 г.

Вариант 24

Цель работы: изучить алгоритм Краскала построения покрывающего леса, научиться использовать его при решении различных задач

Выполнение работы:

№1. Реализовать алгоритм Краскала построения покрывающего леса.

```
int kruskal_algorithm(adjacencyMatrix &g) {
  vector<int> b(g.size());
  for (size_t i = 0; i < g.size(); i++)
    b[i] = i;

  int count = g.size();
  for (size_t i = 0; i < g.size(); i++)
    for (size_t j = 0; j < g.size(); j++)
      if (g[i][j] && b[i] != b[j]) {
        for (size_t k = 0; k < b.size(); k++)
          if (b[k] == b[j])
            b[k] = b[i];
        count--;
      }
  return count;
}
```

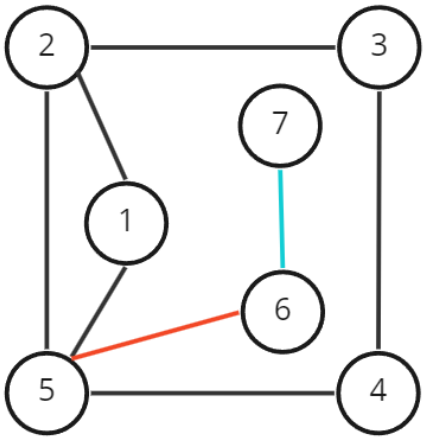
№2. Используя алгоритм Краскала, разработать и реализовать алгоритм решения задачи: «Найти минимальное множество ребер, удаление которых из связного графа делает его несвязным».

```
#include <iostream>
#include <vector>

using namespace std;

using GraphRow = vector<int>;
using Graph = vector<GraphRow>;


int kraskalAlgorithm(Graph &g) {
  vector<int> b(g.size());
  for (size_t i = 0; i < g.size(); i++)
    b[i] = i;

  int count = g.size();
  for (size_t i = 0; i < g.size(); i++)
    for (size_t j = 0; j < g.size(); j++)
      if (g[i][j] && b[i] != b[j]) {
        for (size_t k = 0; k < b.size(); k++)
          if (b[k] == b[j])
            b[k] = b[i];
        count--;
      }
  return count;
}

Graph get_MRows(Graph &g) {
  Graph e(2, (g.size() * g.size() - g.size()) / 2);
  int count = 0;
  for (size_t i = 0; i < g.size(); i++)
    for (size_t j = i; j < g.size(); j++)
      if (g[i][j]) {
        e[0][count] = i;
```

```
        e[1][count] = j;
        cout << '(' << i + 1 << ", " << j + 1 << "): " << count + 1 << endl;
        count++;
       }
  cout << endl;
  for (size_t i = 0; i < 1; ++i)
    e[i].resize(count);

  return e;
}

void outputVector(vector<int> tree) {
  for (auto i: tree)
    cout << i << ' ';
}

void generation_(Graph &g, Graph &e, vector<int> tree,
                 size_t i, int b, bool &isFind) {
  for (size_t x = b; x <= (e.size() - tree.size() + i); x++) {
    tree[i] = x + 1;
    g[e[0][x]][e[1][x]] = 0;
    g[e[1][x]][e[0][x]] = 0;
    if (i == (tree.size() - 1)) {
      if (kraskalAlgorithm(g) == 2) {
        outputVector(tree);
        isFind = true;
      }
    } else
      generation_(g, e, tree, i + 1, x + 1, isFind);

    g[e[0][x]][e[1][x]] = 1;
    g[e[1][x]][e[0][x]] = 1;
  }
}

void generation(Graph &g, Graph &e) {
  bool find = false;
  for (size_t i = 1; !find and i < g.size(); ++i) {
    vector<int> tree(i);
    generation_(g, e, tree, 0, 0, find);
  }
}

int main() {
  Graph v((vector<vector<int >>) {{0, 1, 0, 0, 1, 0, 0},
                                  {1, 0, 1, 0, 1, 0, 0},
                                  {0, 1, 0, 1, 0, 0, 0},
                                  {0, 0, 1, 0, 1, 0, 0},
                                  {1, 1, 0, 1, 0, 1, 0},
                                  {0, 0, 0, 0, 1, 0, 1},
                                  {0, 0, 0, 0, 0, 1, 0}});

  v.output();

  cout << endl;
  auto v_i = get_MRows(v);
  generation(v, v_i);
}
```

№3. Подобрать тестовые данные. Результат представить в виде диаграммы графа

Тестовый граф:



Результат работы программы:

```
C:\BGTU\BGTU\DisMat\lab_4_3\Code\cmake-build-debug\Code.exe
0, 1, 0, 0, 1, 0, 0
1, 0, 1, 0, 1, 0, 0
0, 1, 0, 1, 0, 0, 0
0, 0, 1, 0, 1, 0, 0
1, 1, 0, 1, 0, 1, 0
0, 0, 0, 0, 1, 0, 1
0, 0, 0, 0, 0, 1, 0

(1, 2): 1
(1, 5): 2
(2, 3): 3
(2, 5): 4
(3, 4): 5
(4, 5): 6
(5, 6): 7
(6, 7): 8

{7}
{8}

Process finished with exit code 0
```

Вывод: изучили алгоритм Краскала построения покрывающего леса, научились использовать его при решении различных задач.