Расчётная работа.

Теория графов.

1 Цель работы

1.1 Ознакомиться с понятием графов.

1.2 Узнать какие способы представления графов существуют.

1.3 Научиться решать теоретико-графовые задачи.

2 Задачи

2.1 Придумать алгоритм решения теоретико-графовой задачи.

2.2 Реализовать алгоритм решения задачи на языке программирования C++.

2.3 Протестировать алгоритм.

3 Вариант

3.1 Мой вариант - 2.6. Структура представления графа - неориентированный граф.

4 Список ключевых понятий

4.1 Граф представляет собой пару G = (V, E), где V -множество, элементы которого называются вершинами, а E - набор неупорядоченных пар {v1, v2} из вершин, элементы которых называются ребрами.

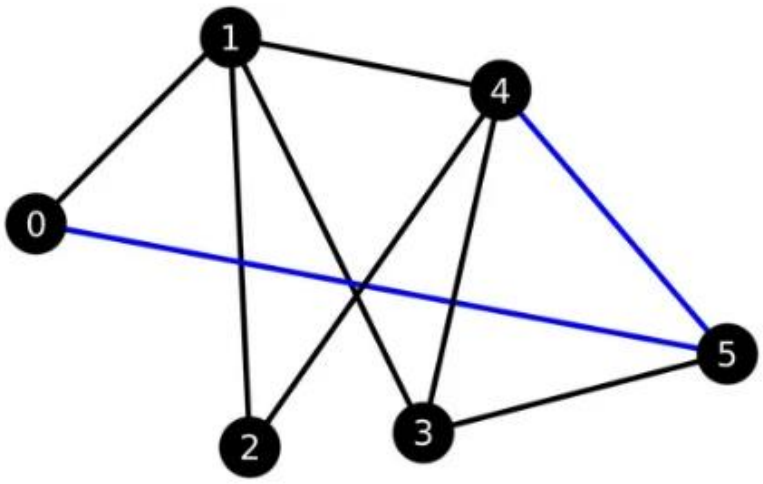

Рисунок 1: Графическое представление графа.

4.2 Путь в графе — последовательность вершин, в которой каждая вершина соединена со следующей ребром.

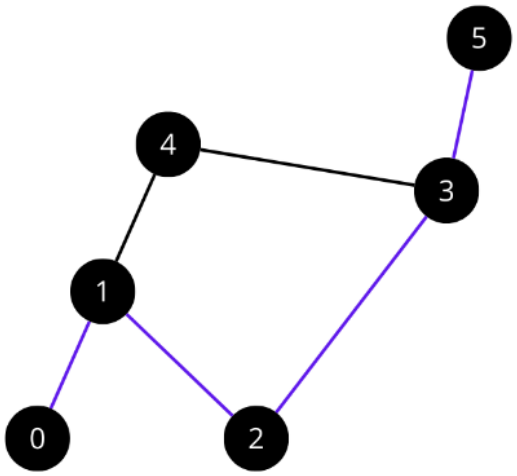

Рисунок 2:Путь в графе.

4.3 Связный граф - граф, в котором существует путь между любыми двумя вершинами.

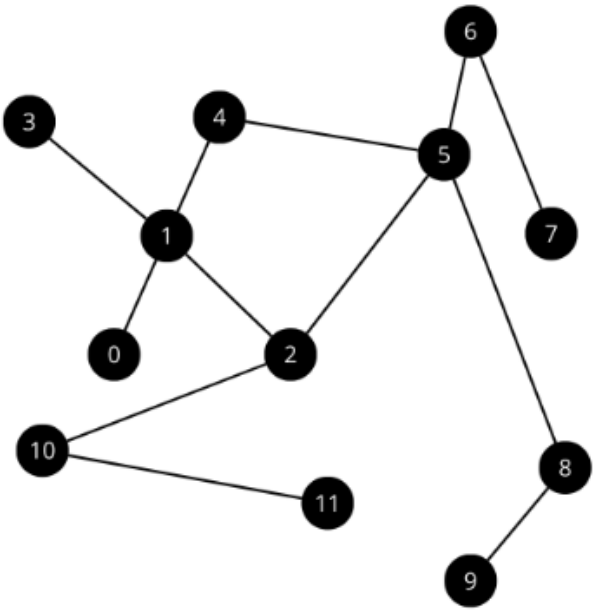

Рисунок 3: Пример связного графа.

4.4 Вершинная связность графа — наименьшее число вершин, удаление которых приводит к несвязному или тривиальному графу.

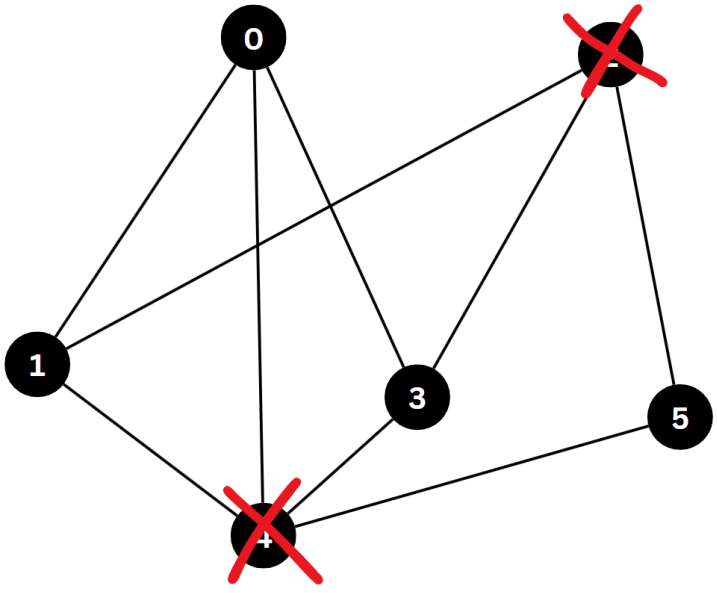

Рисунок 4: Пример вершинной связности равной двум.

5. Описание алгоритма решения

5.1 Проверка связности графа после удаления вершин

Запускаем функцию DFS (обход в глубину) из первой вершины, которая не была удалена. Создаём массив visited, в котором храним, посещена ли вершина. Для каждого вызова DFS проверяем, возможно ли достичь другие вершины графа без удалённых вершин.

5.2 Инициализация графа и удаление вершин

Мы начинаем с пустого множества removed, в которое будем поочередно добавлять вершины для проверки. После удаления вершин, если граф остаётся связанным (т.е. из любой вершины можно добраться до остальных), продолжаем удалять вершины.

5.3 Перебор возможных удалений

Для каждого возможного числа удалённых вершин от 1 до n-1 (где n — количество вершин в графе) мы перебираем все комбинации этих вершин. Для каждой комбинации выполняем DFS, чтобы проверить, остаётся ли граф связанным.

5.4 Проверка несвязанности графа

Если при удалении какой-либо комбинации вершин граф становится несвязанным, возвращаем количество удалённых вершин (это и будет вершинной связностью). Если граф остаётся связанным после удаления любых k вершин, переходим к следующему числу k.

5.5 Завершение работы алгоритма

Если проверка всех возможных удалений не дала результата, возвращаем число n - 1, так как при удалении n-1 вершин граф всегда будет несвязанным.

6. Тестирование программы:

6.1 Первый тестовый пример.

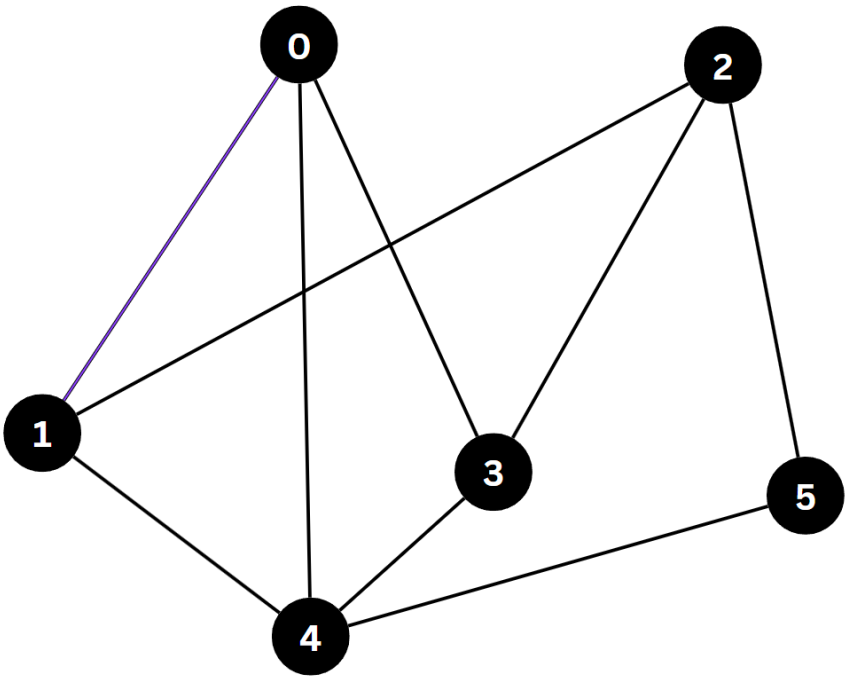

Рисунок 5: Графическое представления тестового графа №1.

При удалении вершин 2 и 4 граф станет несвязанным, следовательно вершинная связанность данного графа равна двум.

Вывод программы:

Вершинная связанность графа: 2

Удаленные вершины: 2 4

6.2 Второй тестовый пример

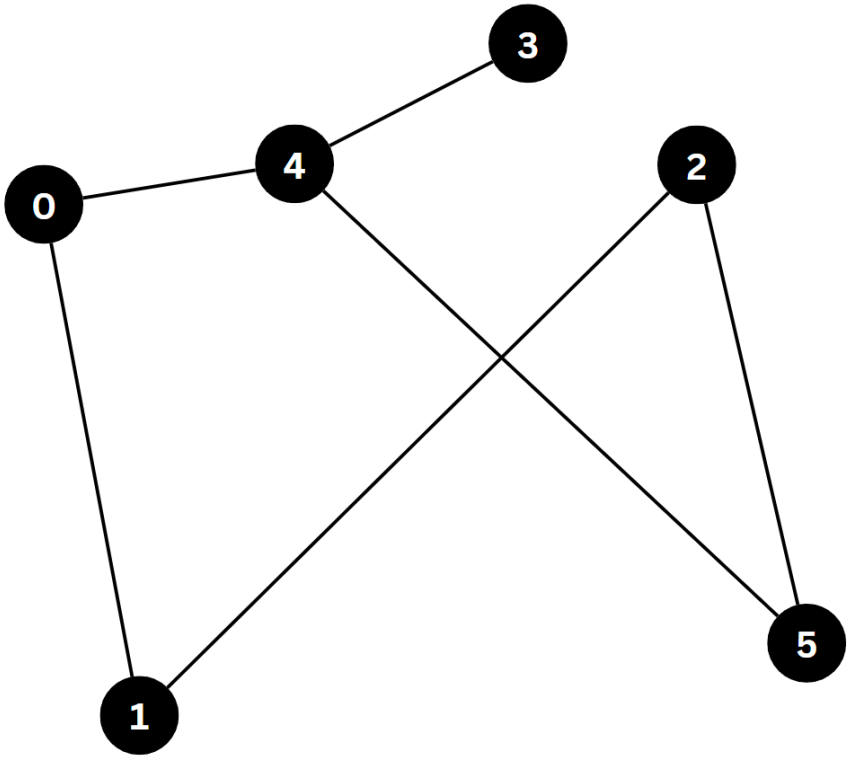

Рисунок 6: Графическое представления тестового графа №2.

При удалении вершины 4 граф станет несвязанным, следовательно вершинная связанность данного графа равна одному.

Вывод программы:

Вершинная связанность графа: 1

Удаленные вершины: 4

6.3 Третий тестовый пример

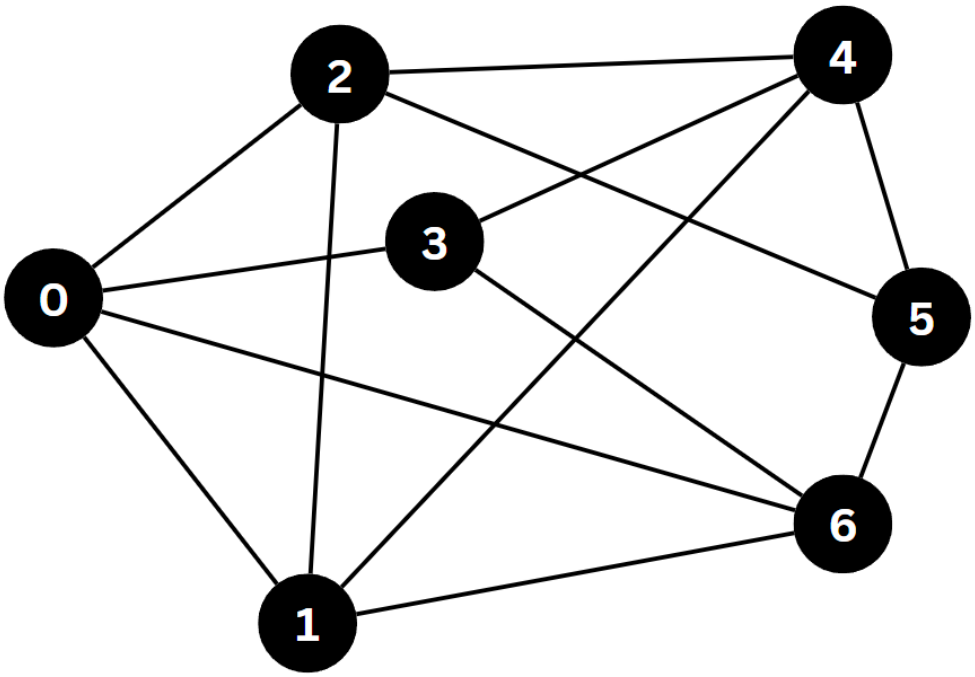

Рисунок 7: Графическое представление тестового графа №3.

При удалении вершин 0, 4 и 6 граф станет несвязанным, следовательно вершинная связанность данного графа равна трем.

Вывод программы:

Вершинная связанность графа: 3

Удаленные вершины: 0 4 6

7. Вывод

В ходе выполнения работы был разработан алгоритм для вычисления вершинной связности графа. Вершинная связность представляет собой минимальное количество вершин, которые нужно удалить из графа, чтобы он стал несвязанным. Алгоритм включает следующие этапы:

1. Построение графа с использованием списка смежности.

2. Проверка связности графа после удаления определенного набора вершин с помощью поиска в глубину (DFS).

3. Перебор всех возможных наборов вершин для удаления, начиная с 1 вершины и до n-1 (где n — количество вершин), с целью нахождения минимального набора, который делает граф несвязанным.

Список литературы

[1] Свободная энциклопедия "Википедия"[Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Граф()

[2] Сайт "MAXimal::algo"[Электронный ресурс]. — Режим доступа:

http://www.e-maxx-ru.1gb.ru/algo/bridgesearching

[3] Сайт "habr"[Электронный ресурс]. - Режим доступа:

https://habr.com/ru/companies/otus/articles/568026/