

Индивидуальное домашнее задание №2

Цель задания

Разработать библиотеку функций, реализующих основные алгоритмы работы с графами. Библиотека должна быть написана на одном из следующих языков программирования: C++, Python, Java. Каждая функция должна быть протестирована на различных тестовых данных, включая граничные случаи.

Дополнительно необходимо провести анализ алгоритмов:

- Измерить время выполнения алгоритмов при больших значениях n (количество вершин или рёбер графа) и представить результаты в виде таблиц или графиков.
- Сравнить алгоритмы по их асимптотической сложности;

Требования к библиотеке

Алгоритмы поиска кратчайшего пути

Реализовать следующие алгоритмы:

- Алгоритм Флойда–Уоршелла;
- Алгоритм Беллмана–Форда.

Алгоритмы обхода графа

Реализовать алгоритмы:

- Поиск в ширину (BFS);
- Поиск в глубину (DFS).

Алгоритмы минимального остовного дерева

Реализовать и сравнить производительность следующих алгоритмов:

- Алгоритм Прима;
- Алгоритм Краскала.

Алгоритмы топологической сортировки

Реализовать:

- Алгоритм Кана;
- Топологическую сортировку через поиск в глубину (DFS).

Алгоритмы поиска сильных компонент связности

Реализовать:

- Алгоритм Тарьяна;
- Алгоритм Косарайю.

Алгоритмы раскраски графов

Реализовать алгоритм раскраски графов с использованием жадного подхода.

Алгоритмы на максимальный поток

Реализовать:

- Алгоритм Форда–Фалкерсона;
- Алгоритм Эдмондса–Карпа.

Требования к реализации

1. Каждая функция должна быть оформлена как отдельная функция или метод в библиотеке.
2. Код должен быть снабжен комментариями, поясняющими основные шаги алгоритма.
3. Все функции должны быть протестированы на следующих типах графов:
 - Ориентированные и неориентированные графы;
 - Связные и несвязные графы;
 - Графы с различными весами рёбер (включая отрицательные веса, где это применимо);
 - Циклические и ациклические графы.
4. Предусмотреть обработку ошибок (например, отсутствие пути между вершинами, невозможность выполнения алгоритма).