Graphs

Release 0.0.1

Wasim

Jul 21, 2020

CONTENTS:

An open-source modern graph (&trees) library built in C++ for exploring graphs with focus on ease of use.

It uses Google Tests for unit testing and Google Benchmark for benchmarking the library. Measure before optimization.

CHAPTER

ONE

DATA STRUCTURES FOR GRAPH REPRESENTATION

The value of a graph library lies in making it easy to construct and analyze graphs.

- adjacencyList
- adjacencyMatrix
- edgeList
- (Coming Up) Multiple formats for printing graphs to make it easy to debug them.

CHAPTER

TWO

ALGORITHMS

- Traversals (BFS, DFS, Level Order Traversal)
- Topological Sort
- Prim's Minimum Spanning Tree
- Kruskals' Minimum Spanning Tree
- Dijsktra's Shortest Path Algorithm (May not work for negative edges)
- Floyd Warshall All Pairs Shortest Path Algorithm
- Bellman Ford Shortest Path Algorithm (Works for negative edges)

CHAPTER

THREE

SAMPLE USAGE

This section shows usage of dijkstra's algorithm on a directed graph. For more usage, see the sample directory directory on github.

```
#include "src/graphs.h"

int main() {
    edgeList edgeList(true); // directed edge list

    edgeList.add_edge(0, 1, 4);
    edgeList.add_edge(0, 2, 2);
    edgeList.add_edge(1, 2, 5);
    edgeList.add_edge(1, 3, 10);
    edgeList.add_edge(2, 4, 3);
    edgeList.add_edge(4, 3, 4);
    edgeList.add_edge(3, 5, 11);

    auto shortest_distance = dijkstra_shortest_distance(edgeList, 0);
    // Expect shortest distance: {0, 4, 2, 9, 5, 20};

    return 0;
}
```

CHAPTER

FOUR

API REFERENCES

Graph Data structures APIs

Graph Algorithm APIs

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search