

Algoritmos e Estruturas de Dados

2º Projeto

Universidade de Aveiro

Diogo Carvalho, Tiago Garcia



Algoritmos e Estruturas de Dados

2º Projeto

Dept. de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

Diogo Carvalho, Tiago Garcia
(113221) diogo.tav.carvalho@ua.pt (114184) tiago.rgarcia@ua.pt

January 3, 2024

Contents

1	Introdução	1
2	GraphTopoSortComputeV1	2
3	GraphTopoSortComputeV2	3
4	GraphTopoSortComputeV3	4
5	Conclusões	5

Chapter 1

Introdução

Neste trabalho será desenvolvido um programa que trabalha com grafos. O programa conseguirá criar ficheiros, quer a partir de funções, quer a partir de um ficheiro de texto. Permitirá ainda realizar a ordenação topológica de um grafo recorrendo a um de três algoritmos implementados, sendo eles:

- 2 através da cópia do grafo e remoção de vértices sem arestas de entrada;
- 3 utilizando uma lista auxiliar com os vértices marcados;
- 4 recorrendo a uma fila de prioridade (first in, first out).

Para cada um destes algoritmos será efetuada a respetiva análise de complexidade. Será ainda feita uma análise comparativa entre os três algoritmos, para determinar qual o mais eficiente.

Os grafos são representados com a seguinte estrutura:

- `int isDigraph` - indica se o grafo é direcionado ou não;
- `int isComplete` - indica se o grafo é completo ou não;
- `int isWeighted` - indica se o grafo tem pesos nas arestas ou não;
- `unsigned int numVertices` - número de vértices do grafo;
- `unsigned int numEdges` - número de arestas do grafo;
- `List *verticesList` - lista de vértices do grafo;

Os vértices são representados com a seguinte estrutura:

- `unsigned int id` - identificador do vértice;
- `unsigned int inDegree` - grau de entrada do vértice;
- `unsigned int outDegree` - grau de saída do vértice;
- `List *edgesList` - lista de arestas do vértice;

As arestas são representadas com a seguinte estrutura:

- `unsigned int adjVertex` - identificador do vértice de destino;
- `double weight` - peso da aresta;

As listas estão implementadas como listas ordenadas, sendo que a ordenação é feita por ordem crescente do identificador do vértice de destino da aresta. A lista está implementada no ficheiro *SortedList.c* e *SortedList.h*.

Chapter 2

GraphTopoSortComputeV1

Chapter 3

GraphTopoSortComputeV2

Chapter 4

GraphTopoSortComputeV3

Chapter 5

Conclusões

Acrónimos