

Disciplina: Teoria dos Grafos

Dupla: Lucas Rolim (114079232) e Anderson Barbosa(114022138).

Decisões de implementação

Linguagem

Para o desenvolvimento da biblioteca foi escolhida a linguagem de programação C++. A escolha dessa linguagem se deu baseada em dois fatores principais: difusão da linguagem e velocidade de processamento de dados e execução. Atualmente C++ é a terceira linguagem de programação mais utilizada no planeta, o que garante uma boa quantidade de fontes para pesquisa e potencial de manutenibilidade. Além disso, em relação a velocidade, C++ é considerada a linguagem de alto nível com melhor potencial de otimização de desempenho por boa parte dos pesquisadores e desenvolvedores.

Paradigma de Programação Orientada a Objeto

Visando garantir uma maior manutenibilidade e também uma maior facilidade desenvolver funcionalidades incrementais nas partes 2 e 3 do trabalho, foi decidido pelo grupo desenvolver a biblioteca utilizando POO. Apesar de perder um pouco em relação à velocidade para o método estrutural esse paradigma de programação permite um maior nível de abstração durante o desenvolvimento, encapsulamento e facilita adições ou mudanças futuras (como peso ou direção nas arestas).

Bibliotecas utilizadas

As seguintes bibliotecas foram utilizadas: `iostream`, `fstream`, `string`, `vector`, `cstdlib`, `queue`, `stack` e `sstream`. O uso de cada uma dessas bibliotecas se deu da seguinte forma:

iostream

Necessária para desenvolver funções que são capazes de exibir determinados valores da estrutura de dados na tela (*cout*).

fstream

Utilizada para escrita e leitura de dados em arquivos de texto. Foi necessária tanto para ler o arquivo contendo as informações sobre o grafo (em que cada linha era uma aresta) quanto para gerar arquivos de saída em formato de texto contendo informações sobre o grafo analisado.

string

Necessária para manipular mais facilmente elementos do tipo string, que era o formato pelo qual se obtinha informações sobre grafo e suas arestas.

vector

Utilizada para construir de maneira otimizada os vetores utilizados ao longo do código, principalmente na parte de criação da estrutura de vetor de adjacência.

queue

Utilizada de modo a aproveitar uma estrutura de fila já pronta e otimizada, disponível para a comunidade de desenvolvedores, de modo a reduzir o tempo de desenvolvimento e execução do programa.

stack

Utilizada de modo a aproveitar uma estrutura de pilha já pronta e otimizada, disponível para a comunidade de desenvolvedores, de modo a reduzir o tempo de desenvolvimento e execução do programa.

sstream

Utilizada para tratar os dados de como os vértices se relacionam. Foi o modo mais eficiente encontrado para transformar uma linha da string do arquivo .txt de entrada em uma informação mais facilmente utilizável para o programa.

Estruturas desenvolvidas

No desenvolvimento do algoritmo foram utilizadas as estruturas de fila e pilha para a implementação das operações com o grafo. Essas estruturas foram utilizadas por se encaixarem naturalmente na natureza dos problemas e realizarem operações em tempo $O(1)$ nesse contexto.

Em relação ao grafo, foram implementadas duas estruturas de representação: matriz de adjacência e vetor de adjacência. A estrutura matriz de adjacência foi representada por um ponteiro de ponteiro para booleanos, o que reduziu o espaço de memória gasto e aumentou a velocidade dessa estrutura. O uso de boolean, por exemplo, poupou 3 bytes de memória a cada elementos da matriz e diminuiu em quatro vezes o espaço de memória necessário para estrutura caso essa armazenasse inteiros.

A estrutura vetor de adjacência, por sua vez, foi criada utilizando um vetor de vetor para inteiros e fazendo uso da biblioteca std vector. O uso da biblioteca std::vector no vetor de adjacência permitiu o instanciamento de objetos antes de saber seus valores finais de modo a não gerar erros e com o máximo de otimização.

Ambiente de testes

Sistema Operacional: Microsoft Windows 7

Arquitetura: 64 bits

Configuração do Computador: Intel i5 (terceira geração) e 8 Megabytes de memória RAM.