Trabalho 1

Métodos de Programação

Lincoln Abreu Barbosa, matrícula 14/0045023

# Descrição da demanda

O Trabalho 1 é constituído de 3 partes essenciais: a biblioteca de grafos (grafo.c), a biblioteca de testes (testa\_grafo.c) com Makefile e este documento de descrição das etapas.

A biblioteca de Grafos deve ser constituída das seguintes funções:

cria\_grafo (nome): retorna estrutura de dados grafo que tem um nome

retorna\_nome\_grafo(G): retorna o nome do grafo G

destroi\_grafo(G): destrói grafo g

adjacente(G, x, y): verifica se x e y são adjacentes em G

vizinhos(G, x): retorna lista de todos os vértices que podem ser visitados a partir de x (existe uma aresta que vai de x para este vértice)

adiciona\_vertice(G, x): adiciona vértice x no grafo G

remove\_vertice(G, x): remove vértice x do grafo G

adiciona\_aresta(G, x, y): no grafo G adiciona uma aresta de x para y

remove\_aresta(G, x, y): remove a aresta que vai de x para y

retorna\_valor\_vertice(G, x): retorna o valor associado com o vértice x

muda\_valor\_vertice(G, x, v): coloca valor v no vértice x em G

retorna\_valor\_aresta(G, x, y): retorna valor da aresta x,y em G

muda\_valor\_aresta(G, x, y, v): coloca o valor v na aresta x,y em G

# Planejamento

O planejamento se caracterizou pela pesquisa a respeito da estrutura de dados de Grafos e no reuso de material e código utilizados em outras disciplinas. Todos os códigos utilizados foram produzidos pelo próprio autor com auxílio de material complementar fornecido pela disciplina de Métodos de Programação ou por disciplinas cursadas anteriormente, tais como livros-texto ou códigos exemplos, a fim de auxiliar na estruturação lógica.

# Codificação da biblioteca de grafos

A biblioteca de grafos foi codificada utilizando pseudocódigo e depois sendo transcrita para código C.

Foi produzida em ambiente linux e respeitando as normas ANSI, cada função foi descrita de forma completa para depois ser corrigida conforme auxilio das mensagens do próprio GCC e eventuais falhas de lógica foram diagnosticadas utilizando a ferramenta GDB, disponíveis naturalmente no ambiente Linux (ubuntu 14.04).

# Codificação da biblioteca de testes

A biblioteca de testes foi produzida utilizando a biblioteca de grafos como embasamento, portanto é constituída por:

Int Testa\_cria\_grafo (): Aciona a função cria\_grafo repassando a string “teste” como parâmetro, checa se o retorno da função é um tipo grafo e aciona a função retorna\_nome\_grafo passando como parâmetro o grafo criado na primeira situação, certificando-se de que o retorno da segunda função é uma string “teste”. Retorna 1 para teste com sucesso e 0 para falha.

Int Teste\_adiciona\_vertice(): Aciona a função cria\_grafo repassando a string “teste” como parâmetro para então executar adiciona\_vertice com os parâmetros : grafo retornado por cria\_grafo e o inteiro

“3”. Retorna 1 ao certificar que grafo->fistno->índice é “3”, 0 para o contrário.

Demais testes a serem implementados.

##### Bibliografia

Modelo de artigo fornecido pelo IEEE no link:

[trans\_jour.docx](https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwii67S_ooHPAhWKFpAKHWMaCpUQFggsMAE&url=https%3A%2F%2Fwww.ieee.org%2Fdocuments%2Ftrans_jour.docx&usg=AFQjCNG5Zs_A5J9zxhPewVA3-5IFV5Eh5A)

Material da disciplina disponível em:

http://aprender.ead.unb.br/pluginfile.php/267692/mod\_forum/attachment/212774/MP\_Trab\_1.pdf