

Universidade de Brasília
Departamento de Ciência da Computação
Disciplina: Métodos de Programação
Código da Disciplina: 201600

Métodos de Programação - 201600

Trabalho 1

Crie uma biblioteca de grafo direcionado em C ou C++ com as seguintes funções:

Para cada função se deve especificar qual o tipo de retorno, dos parâmetros e o que acontece no caso de haver erro.

`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

O G é um ponteiro para grafo. Os identificadores dos vértices são valores inteiros. Os valores são ponteiros para estruturas. Estas estruturas podem conter diversos dados.

Faça um programa executável chamado `testa_grafo.c` (ou `.cpp`) que utiliza o módulo `grafo.c` que implementa um grafo usando a interface definida em `grafo.h` (ou `.hpp`).

O programa `testa_grafo.c` (ou `.cpp`) deve testar se a implementação do grafo funciona corretamente. Devem ser tratadas as exceções. Ex: O que acontece quando se tenta destruir um grafo que não existe?

O programa e o módulo devem ser depurados utilizando o GDB.

(<http://heather.cs.ucdavis.edu/~matloff/UnixAndC/CLanguage/Debug.html>)

(<https://www.cs.umd.edu/~srhuang/teaching/cmsc212/gdb-tutorial-handout.pdf>)

Faça um Makefile utilizando o exemplo de makefile 5 dado em :
(<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>)

Deve ser gerado um documento (pdf ou odt) que diz como cada função foi testada. Para cada caso deve constar:

- 1) Nome da função, parâmetros e significado dos parâmetros. Especificação da função
- 2) Para cada um dos testes em cada função
 - 2.1) Nome de cada teste
 - 2.2) O que vai ser testado
 - 2.3) Qual deve ser a entrada
 - 2.4) Qual deve ser a saída
 - 2.5) Qual é o critério para passar no teste
 - 2.6) Se a sua função efetivamente passou no teste ou não
- 3) Responda em que casos a função não retorna um resultado válido.
 - 3.1) Existem funções que podem corromper a estrutura de dados? Como?
 - 3.2) O que pode ser feito para evitar este problema

Devem ser enviados para a tarefa no ead.unb.br um arquivo zip onde estão compactados todos os diretórios e arquivos necessários. Todos os arquivos devem ser enviados compactados em um único arquivo (.zip) e deve ser no formato matricula_primeiro_nome.zip. ex: 06_12345_Jose.zip. Deve ser enviado um arquivo dizendo como o programa deve ser rodado.

Data de entrega:

5/4/17

Pela tarefa na página da disciplina no ead.unb.br