

Trabalho #Final - Aplicações com Grafos

- **Data para entrega:** 02/07/2018 (até 23h55), via moodle.

Implementação individual.

Em caso de **cópia** de código, **os alunos envolvidos** terão nota igual a **zero** no trabalho final e nas avaliações restantes da disciplina (P2 e APS).

Objetivo:

- Implementar um programa na linguagem C com as seguintes características:
 - * Construa um grafo não direcionado e **ponderado** (com peso nas arestas) usando **listas de adjacências** como estrutura de dados para representar o grafo;
 - * O sistema deverá ter uma interface (menu) com as seguintes opções:
 - Inserir Vértice
 - Inserir Aresta
 - Remover Vértice
 - Remover Aresta
 - Visualizar Grafo
 - Informar grau de um vértice
 - Informar se o grafo é conexo
 - Converter grafo para Matriz de Adjacência
 - Caminhamento em Amplitude (Busca em Largura)
 - Caminhamento em Profundidade
 - Caminho Mínimo (Algoritmo de Dijkstra)
 - Árvore Geradora Mínima (Algoritmo de Prim)

Requisitos:

- O identificador dos nós (vértices) do grafo deverá ser do tipo **int**;
- As arestas deverão ter custos (positivos e não negativos) para que possam ser encontrados os caminhos mínimos e árvore geradora mínima.

Avaliação:

O trabalho será avaliado em função da:

1. Correção (o sistema cumpre com as exigências);
2. Documentação (o sistema está devidamente comentado);
3. Modularidade (o sistema está bem estruturado, com funções parametrizadas);
4. Robustez (o sistema não trava em tempo de execução).

5. Desenvolvimento em aula ao longo do bimestre.

- Detalhamento dos itens que serão avaliados:

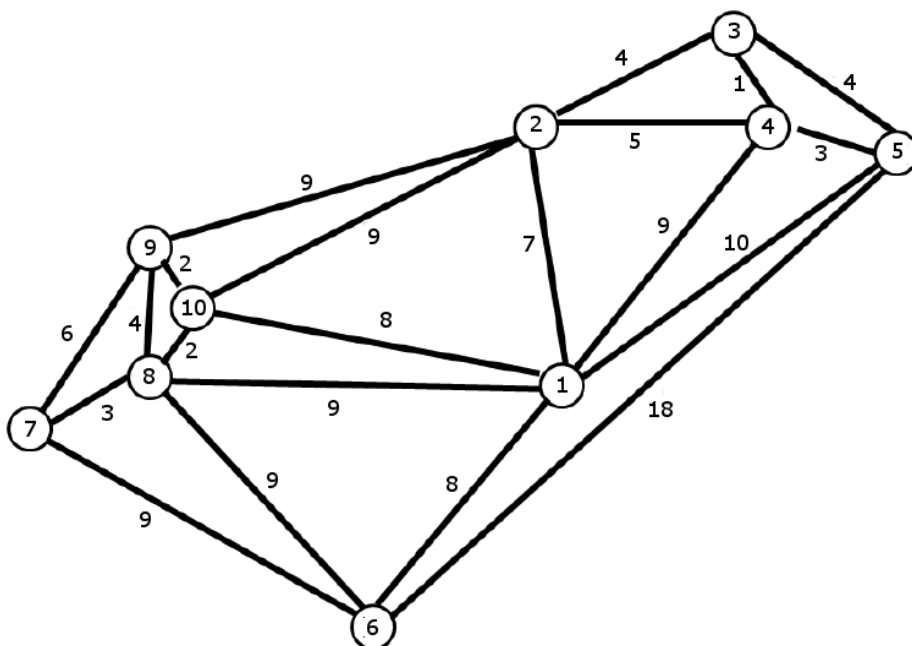
- Utilizou estrutura de dados em listas de adjacências (alocação dinâmica de memória) para representar o grafo?
- Liberou a memória alocada?
- Inseriu e removeu vértices no grafo corretamente? Permitiu a inserção de vértices com identificação repetida?
- Inseriu e removeu arestas do grafo corretamente? Permitiu a inserção de arestas repetidas?
- Visualiza-se de forma clara e correta o grafo?
- A entrada de dados e saída de dados é amigável?
- Implementou funções parametrizadas?

Ex. `inserirVertice(vert), inserirAresta(vert1,vert2).`

- O programa travou em tempo de execução?
- O programa está legível (identado) e bem documentado (comentado)?
- O programa permite a inclusão, **via código**, de um grafo teste (utilizando as funções parametrizadas)?

Forma de entrega: via moodle da disciplina. O prazo da data e hora de entrega não será prorrogado. A não submissão do trabalho final implica em nota zero para o mesmo.

Um exemplo de grafo a ser utilizado para teste:



Resultado da Busca em Largura com vértice origem igual a 1:

v (vértice)	$\pi[v]$ (pai do vértice)	$d[v]$ (distância da origem)
1	-1	0
2	1	1
3	2	2
4	1	1
5	1	1
6	1	1
7	6	2
8	1	1
9	2	2
10	1	1

Resultado da Busca em Profundidade:

v (vértice)	$d[v]$ (descoberta)	$f[v]$ (finalização)	$\pi[v]$ (pai do vértice)
1	1	20	-1
2	2	19	1
3	3	18	2
4	4	17	3
5	5	16	4
6	6	15	5
7	7	14	6
8	8	13	7
9	9	12	8
10	10	11	9

Árvore Geradora Mínima pelo algoritmo de Prim:

v (vértice)	$\pi[v]$ (pai do vértice)	$key[v]$
1	-1	0
2	1	7
3	2	4
4	3	1
5	4	3
6	1	8
7	8	3
8	10	2
9	10	2
10	1	8

Árvore de Caminhos Mínimos pelo algoritmo de Dijkstra:

v (vértice)	$\pi[v]$ (pai do vértice)	$d[v]$
1	-1	0
2	1	7
3	4	10
4	1	9
5	1	10
6	1	8
7	8	12
8	1	9
9	10	10
10	1	8