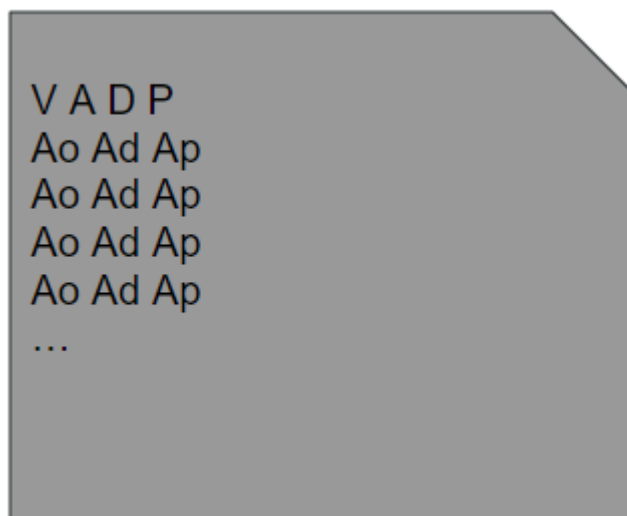


# Trabalho M1 - Parte 2

A parte 2 consiste na implementação da leitura de arquivos de grafos e na implementação de 3 algoritmos de busca e a leitura do Grafo por um arquivo de texto.

O formato do arquivo a ser lido deve seguir o esquema apresentado nos slides:



```
V A D P
Ao Ad Ap
Ao Ad Ap
Ao Ad Ap
Ao Ad Ap
...
```

Onde:

V = Número de vértices

A = Número de Arestas

D = Direcionado ou não (0 para não e 1 para sim)

P = Ponderado ou não (0 para não e 1 para sim)

A primeira linha é sempre presente, seguida de A linhas, uma linha para cada aresta, onde:

Ao = Vértice de origem da aresta

Ad = Vértice de destino da aresta

Ap = Peso da aresta (somente presente em grafos ponderados)

Lembrando que em grafos não direcionados as arestas devem ser criadas nos dois sentidos

Quando um grafo é criado ele pode receber um arquivo de texto nesse formato e criar os vértices e arestas a partir de seu conteúdo.

Os algoritmos implementados serão os seguintes:

**Busca em largura** - Dando um vértice de origem do grafo deve ser impressa a ordem em que os vértices são visitados até que todos os vértices alcançáveis sejam impressos.

**Busca em profundidade** - Similar a busca em largura, dando um vértice de origem do grafo deve ser impressa a ordem em que os vértices são visitados até que todos os vértices alcançáveis sejam impressos.

E por último, o algoritmo mais importante:

**Dijkstra** - Dado um grafo ponderado e um vértice de origem, ele deve retornar e imprimir uma lista da menor distância do vértice de origem para cada outro vértice do grafo e a ordem de vértices do caminho da origem até cada um desses vértices.

Importante:

- Os três algoritmos devem executar tanto na lista quanto na matriz de adjacência com o mesmo código.
- A entrega do trabalho completo (parte 1 e 2) será no dia **12/09**.