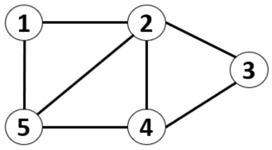
Introdução à Teoria dos Grafos

1. Representação Gráfica:



$G = \{V(G), E(G)\}$

2. Conjuntos:

$V(G)$: conjunto de vértices (entidades)

$$V(G) = \{1,2,3,4,5\}$$

$E(G)$: conjunto de arestas (relações entre vértices)

$$E(G) = \{(1,2),(1,5),(2,3),(2,4),(2,5),(3,4),(4,5)\}$$

3. Dimensões:

$v(G) = |V(G)| = n \;\; \Rightarrow$ ordem

$e(G) = |E(G)| = m$

tamanho = *n+m*

4. Conceitos Básicos:

- Ψ_e: extremos de uma aresta *e* (vértices incidentes de *e*)
- Vértices adjacentes: dois vértices são adjacentes se forem incidentes da mesma aresta;

  Arestas: $(2,4)$ e $(2,5)$

- Arestas paralelas: seja e_1 e e_2 arestas paralelas, então $\Psi_{e_1} = \Psi_{e_2}$

- Loop: aresta cujos extremos são iguais
- Grafo simples: grafo que não possui loops ou arestas paralelas
- Multigrafo: loop e arestas paralelas fazem parte do grafo (grafo não simples)
- Vizinhança do vértice v ($N_{(v)}$) : conjunto de vértices que possui relação (aresta) com v.

  $N_{(3)} = \{2,4\}$

- Vizinhança fechada do vértice v ($N_{[v]}$) : $N_{[v]} = N_{(v)} \cup \{v\}$

  $N_{[1]} = \{1,2,5\}$

- Vértice universal: v é universal quando $N_{[v]} = V(G)$.

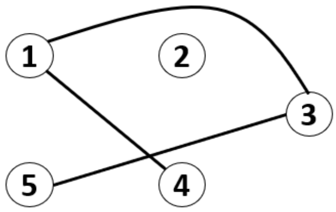
  Vértice 2 é universal

- Complemento de G: sumponha G um grafo simples

$$\bar{G} = \{V(G), \bar{E}(G)\}$$

  onde

$$\bar{E}(G) = \{(u, v) \mid (u, v) \notin E(G), u \neq v\}$$



$\bar{G} = \{V(G), E(G)\}$

- O crescimento das arestas se dá na ordem quadrática ($O(n^2)$)

$$|E(G)| + |\bar{E}(G)| = \binom{n}{2} = \frac{n(n-1)}{2} \quad O(n^2)$$

- Distância entre dois vértices v e w ($d(v, w)$): tamanho do caminho entre dois vértices com a menor quantidade de arestas.

  $d(2,5) = 1$

  $d(1,4) = 2$

5. Representação de grafos

Existem duas maneiras padrão para representar um grafo $G = (V, E)$: lista de adjacências e matriz de adjacência.

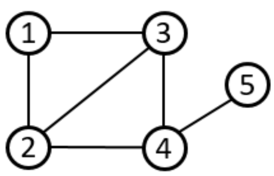
5.1. Lista de Adjacência

Consiste em um arranjo Adj de $|V|$ listas, uma para cada vértice em V. Para cada $u \in V$, a alista adjacente $Adj[u]$ consiste em todos os vértices v tais que existe uma aresta $(u, v) \in E$

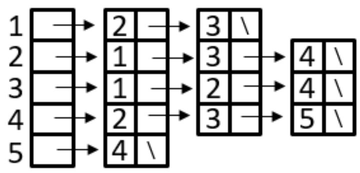
5.2. Matriz de Adjacência

Supomos que os vértices são numerados $1,2,3,..,|V|$ de alguma maneira arbitrária. Então, a matriz de adjacência consiste em uma matriz $|V|x|V|$ $A(a_{ij})$ tal que

$$a_{ij} = \begin{cases} 1 & \text{se } (i,j) \in E, \\ 0 & em\ caso\ contrário \end{cases}$$



$G = (V, E)$



Adj

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	1	0
3	1	1	0	1	0
4	0	1	1	0	1
5	0	0	0	1	0

$A = (a_{ij})$

6. Algoritmos de busca em grafos

Seja G o grafo de entrada, $s \in V[G]$ o vértice origem da busca, e os arranjos *cor[v]*, *d[v]* e $\pi[v]$ representando respectivamente a cor, a distância da origem e o predecessor na pesquisa do vértice v. Segue os algoritmos básicos de busca em grafo.

6.1. Busca em largura (extensão)

```
BFS (G, s)
for cada vértice de u ∈ V[G] − {s}
    do cor[u] ← BRANCO
       d[u] ← ∞
       π[u] ← NULL
cor[s] ← CINZA
d[s] ← 0
π[s] ← NULL
Q ← ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    do u ← DEQUEUE(Q)
       for cada vértice de v ∈ adj[u]
           do if cor[v] = BRANCO
              then cor[v] ← CINZA
                   d[v] ← d[v] + 1
                   π[v] ← u
                   ENQUEUE(Q, v)
           cor[u] ← PRETO
```

7.2. Busca em profundidade

```
DFS (G)
for cada vértice de u ∈ V[G]
    do cor[u] ← BRANCO
       π[u] ← NULL
for cada vértice de u ∈ V[G]
    do if cor[u] = BRANCO
       then DFS − VISIT(u)
```

```
DFS − VISIT (u)
cor[u] ← CINZA
for cada vértice v ∈ adj[u]
    do if cor[u] = BRANCO
        then π[v] ← u
        DFS − VISIT(v)
cor[u] ← PRETO
```

Atividade de implementação:

1. Implemente os algoritmos de busca em grafos.
    a) Implemente as estruturas: matriz de adjacência e lista de adjacência
    b) Implemente um leitor de grafos em arquivos
    c) Implemente Busca em largura
    d) Implemente Busca em Profundidade