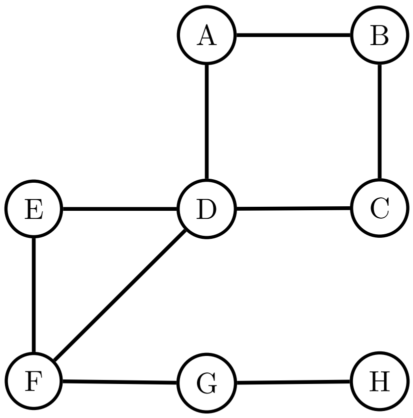
# Aula Prática 8

ASA 2024/2025

**Q1 (T1 06/07 I.1)** Considere o grafo não dirigido seguinte.



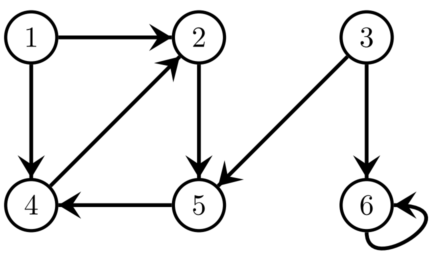

Para todos os nós $u$, calcule $d[u]$ e $\pi[u]$ obtidos por uma procura em largura primeiro a partir do nó $A$.

**Solução:**

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| d | 0 | 1 | 2 | 1 | 2 | 2 | 3 | 4 |
| $\pi$ | - | A | B | A | D | D | F | G |

Q: A B D C E F G H

**Q2 (Ex. 22.2-1)** Aplique uma BFS no seguinte grafo, a começar em 3 e utilizando a ordem númerica para os vizinhos. Indique também o $d$ e $\pi$ de cada valor.




**Solução:**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| d | $\infty$ | 3 | 0 | 2 | 1 | 1 |
| $\pi$ | - | 4 | - | 5 | 3 | 3 |

Q: 3 5 6 4 2

**Q3 (Ex. 22.2-7)** Proponha um algoritmo $O(V + E)$ que reconheça se um dado grafo é bipartido. Produza a partição ou mostre que tal partição não existe.

**Solução:**

1. Usar uma BFS considerando: os níveis pares no conjunto de vértices L, e os níveis ímpares no conjunto de vértices R
2. Validar os arcos para trás/frente/cruzamento. Se $d[u]\%2$ da visita inicial ao nó $u$ fôr diferente do valor $(d[v] + 1]\%2$ quando visitamos $u$ numa vez subsequente, atravessando o arco $(v, u)$, então não é bipartido!

Observações:

- Equivalente ao problema 2-coloring clássico
- É irrelevante se o grafo $G$ é dirigido ou não. Podemos considerar como não dirigido

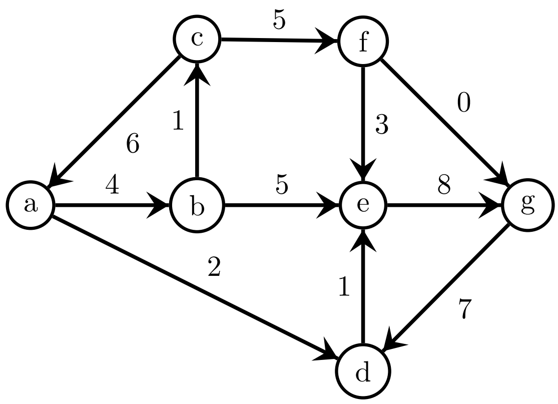
**Q4 (R1 08/09 I.3)** Considere a aplicação de uma pesquisa em largura (BFS) num grafo $G = (V, E)$, onde $s \in V$ é o vértice origem da BFS. Considere ainda a utilização da árvore BF para classificação dos arcos, tal como na execução de uma pesquisa em profundidade (DFS). Assim, para cada uma das seguintes afirmações, indique se é verdadeira (V) ou falsa (F).

1. A BFS permite identificar os caminhos mais curtos para todos os vértices do grafo atingíveis a partir de $s$.
2. Sejam $u$ e $v$ vértices do grafo atingíveis a partir de $s$ tal que $d[v] > d[u] + 1$. Nesse caso, o arco $(u, v)$ não existe no grafo.
3. Se o grafo $G$ for não dirigido, na aplicação de uma BFS podem existir arcos para a frente.
4. Sejam $u$ e $v$ dois vértices do grafo atingíveis a partir de $s$ tal que $d[v] > d[u]$. Então temos necessariamente que $d[v] - d[u]$ denota o número de arcos no caminho mais curto de $u$ para $v$.
5. Para cada arco $(u, v)$ da árvore BF temos que $d[v] = d[u] + 1$.
6. Se o grafo $G$ for não dirigido, na aplicação de uma BFS não existem arcos de cruzamento.

**Solução:**

| 1. | 2. | 3. | 4. | 5. | 6. |
|---|---|---|---|---|---|
| V | V | F | F | V | F |

**Q5 (T1 06/07 II.1)** Considere o grafo da figura.

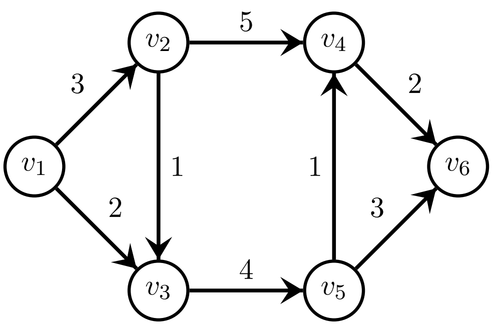



Indique os valores de $d$ e $\pi$ para cada vértice quando faltam extrair dois nós da fila de prioridade na execução do algoritmo de Dijkstra a partir do vértice $c$.

**Solução:**

| | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| d | 6 | 10 | 0 | 8 | 8 | 5 | 5 |
| $\pi$ | c | a | Nil | a | f | c | f |

**Q6 (T1 08/09 II.3)** Considere o grafo da figura.




1. Indique os valores de $d$ e $\pi$ para cada vértice imediatamente após a aplicação do procedimento Relax sobre todos os arcos com origem no vértice $v_5$, durante a execução algoritmo de Dijkstra a partir do vértice $v_1$.

2. Indique os valores de $d$ e $\pi$ para cada vértice imediatamente após serem processados 5 arcos durante a execução do algoritmo para cálculo de caminhos mais curtos de origem única em grafos dirigidos acíclicos (DAGs) a partir do vértice $v_1$. Deve considerar a ordenação topológica mais pequena por ordem lexicográfica.

**Solução:**

**a)**

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| d | 0 | 3 | 2 | 7 | 6 | 9 |
| $\pi$ | Nil | $v_1$ | $v_1$ | $v_5$ | $v_3$ | $v_5$ |

**b)** Ordenação topológica: $\langle v_1, v_2, v_3, v_5, v_4, v_6 \rangle$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| d | 0 | 3 | 2 | 8 | 6 | $\infty$ |
| $\pi$ | Nil | $v_1$ | $v_1$ | $v_2$ | $v_3$ | Nil |

**Q7 (CLRS Ex. 24.3-3)** Suponha que alteramos a linha 4 do algoritmo de Dijktra da seguinte forma:

**while** $|Q| > 1$

Isto resulta no ciclo **while** ser executado $|V| - 1$ vezes em vez de $|V|$ vezes. A alteração proposta está correcta?

**Solução:**

Sim, está correcto, dado que não podem existir pesos negativos:

- se $u$ não é alcançável, $d[u] = \delta(s, u) = \infty$
- se $u$ é alcançável, então existe um caminho $p = s \rightsquigarrow x \to u$ e quando $x$ passou de $Q$ para $S$ o arco $(x, u)$ foi relaxado. Logo, $d[u] = \delta(s, u)$.

**Q8 (CLRS Ex. 24.3-4)** O Professor Gaedel escreveu um programa que ele alega ser a implementação do algoritmo de Dijkstra. O programa produz `d[v]` e $\pi$[`v`] para cada vértice $v \in V$. Indique um algoritmo que execute em tempo $O(V + E)$ para verificar o output do programa do professor. O seu algoritmo deve determinar se os valores em $d$ e $\pi$ correspondem a uma árvore de caminhos mais curtos. Pode assumir que todos os pesos dos arcos são não-negativos.

**Solução:**

Verificar que:

1. o resultado de $\pi[v]$ é uma árvore
2. $d[s] == 0$ && $\pi[s] == NIL$
3. $d[v] == d[\pi[v]] + w(\pi[v], v), \forall v \neq s$
4. $d[v] == \infty$ se e só se $\pi[v] == NIL, \forall v \neq s$

Se alguns deste passos falhar, **não é Dijkstra**

Se não falhar, confirmar fazendo uma passagem de relaxação em todos os arcos:

- se algum arco mudar, **não é Dijkstra**
- se não mudar, o resultado **é correcto**