

Universidade Federal de Roraima
Departamento de Ciência da Computação
Análise de Algoritmos

2ª Lista

Aluno: Rodrigo dos Santos Tavares

Questão 2:

$$\text{a)} \quad T(n) = \begin{cases} c, & n = 1 \\ aT\left(\frac{n}{b}\right) + c, & n > 1 \end{cases}$$

$$\begin{aligned} T(n) &= a \left[aT\left(\frac{n}{b^3}\right) + c \right] + ac + c = a^2 T\left(\frac{n}{b^2}\right) + a^1 c + a^0 c \\ &= a^2 \left[aT\left(\frac{n}{b^3}\right) + c \right] + ac + c = a^3 T\left(\frac{n}{b^3}\right) + a^2 c + a^1 c + a^0 c \end{aligned}$$

$$a^k T\left(\frac{n}{b^k}\right) + \sum_{i=0}^{k-1} a^i c$$

$$n = b^k, \quad k = \log_b n$$

$$\begin{aligned} &a^k T(1) + \sum_{i=0}^{k-1} a^i c \\ &c \left(\frac{a^{(\log_b n)+1} - 1}{a - 1} \right) \end{aligned}$$

$$\text{b)} \quad T(n) = \begin{cases} \Theta(1), & n = 1 \\ 2T\left(\frac{n}{2}\right) + \Theta(n), & n > 1 \end{cases}$$

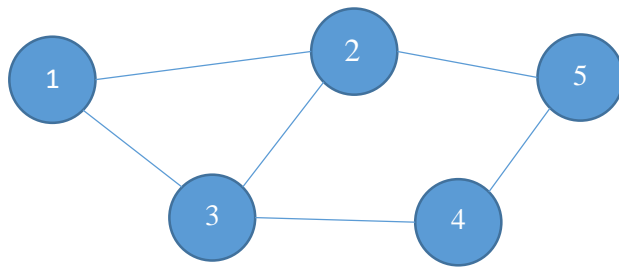
Método mestre:

$$n^{\log_2 2} = n = \Theta(n)$$

$$T(n) = \Theta(n \log n)$$

Questão 3:

- a) Grafos são estruturas usadas para representar relações entre objetos, sendo representados por $G(V, E)$, em que V é o conjunto de vértices, e E é o conjunto de arestas. Por exemplo, o Grafo $G(V, E) \mid V = \{1, 2, 3, 4, 5\}, E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}$, pode ser representado da seguinte forma:



b) **Grafos conexos** são grafos em que existe um caminho entre todos os pares de vértices.

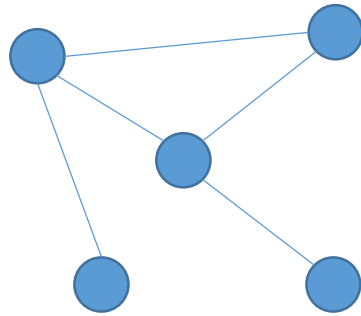


Figura 1: Exemplo de grafo conexo

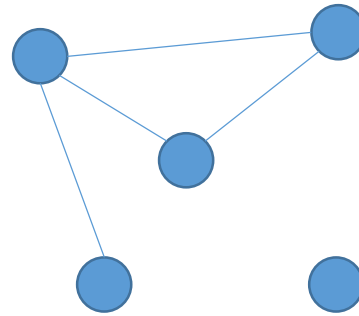
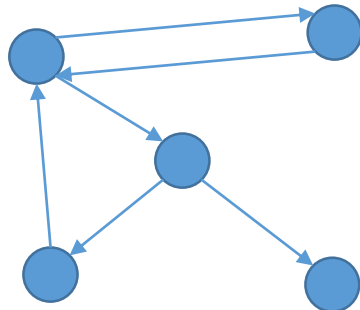


Figura 2: Exemplo de grafo desconexo

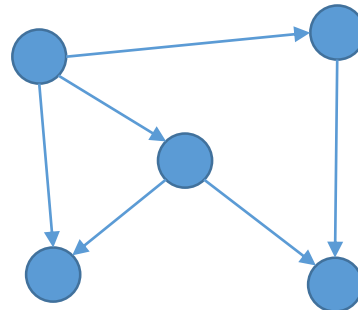
A figura 3 representa um grafo desconexo, pois existe um vértice *inalcançável*.

Grafos acíclicos são grafos cujos caminhos não repetem vértices.

Grafos direcionados são grafos cujas arestas têm uma direção, ou seja, partem de um vértice em direção a outro.

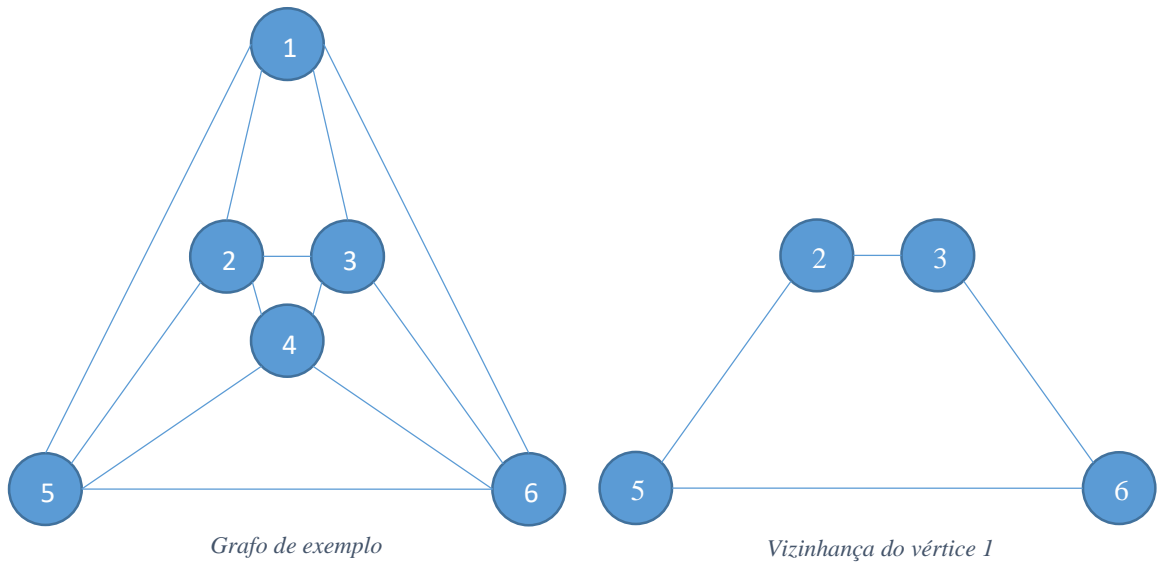


Exemplo de grafo direcionado

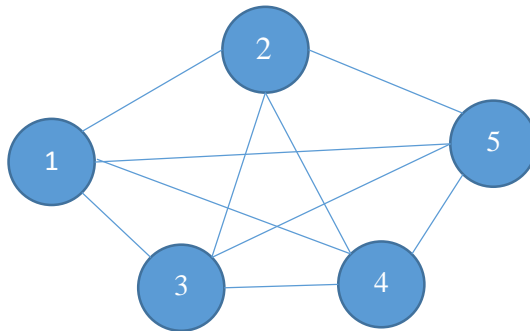


Exemplo de grafo acíclico

c) Em um grafo, dois vértices são **adjacentes** quando têm uma aresta entre eles. Já a **vizinhança** de um vértice é o grafo formado pelos seus adjacentes, mantendo-se suas arestas.

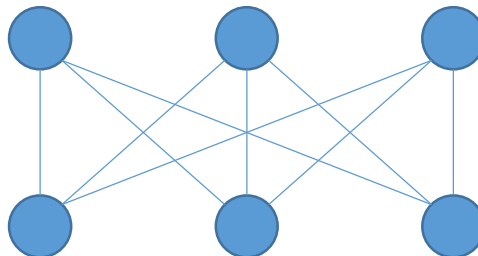


- d) Um **grafo planar** é um grafo que pode ser desenhado no plano sem que nenhuma de suas arestas se cruzem.
- f) Um **grafo completo** é um grafo em que todos os vértices são adjacentes uns aos outros.

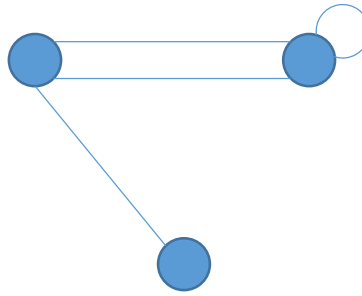


Clique é um subconjunto de um grafo em que seus vértices e arestas formam um grafo completo.

Um **grafo bipartido** consiste em um grafo em que, dividindo-se o grafo em dois conjuntos U e V de vértices, todas as arestas do grafo conectam vértices do conjunto U com o conjunto V .



- g) Um **grafo simples** é um grafo que não tem laços e nem mais de uma aresta ligando dois vértices.
- Chama-se **multigrafo** um grafo que não é simples, ou seja, pode ter laços e mais de uma aresta ligando dois vértices.

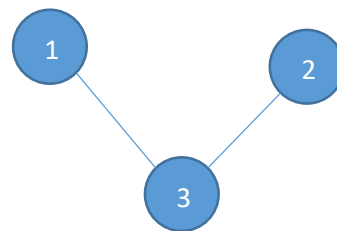


Um **digrafo** é um grafo direcionado.

Questão 4:

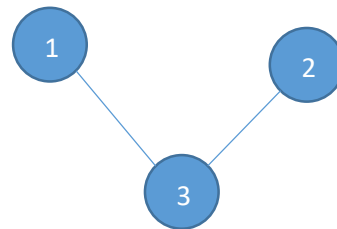
Matriz de incidência é uma matriz bidirecional que representa a relação entre vértices e arestas.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$



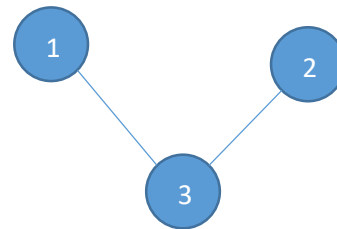
Matriz de adjacência é uma matriz que representa relação de adjacência entre vértices.

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



Uma **lista de adjacências** consiste em um conjunto de V listas de vértices adjacentes a cada vértice do grafo.

$$\{(1, \{3\}), (2, \{3\}), (3, \{1, 2\})\}$$



Questão 5:

Uma tabela hash é uma associação entre chaves e valores. As chaves, quando usadas como argumento de uma função *hash*, retornam um valor da tabela, o que permite que se possa fazer uma “busca” com complexidade $O(1)$.

Uma função *hash* pode acabar gerando o mesmo resultado para chaves diferentes, causando conflito na tabela. Uma maneira de resolver este conflito seria usando uma lista de valores no lugar de apenas um valor, o que aumentaria a complexidade das operações para $O(n)$.

Como alternativa à lista, pode-se usar também árvores balanceadas para os valores das chaves, garantindo uma menor complexidade $O(\log n)$.

Questão 6:

No **método da enumeração**, se enumera as soluções para um problema. Enquanto na **enumeração explícita** se enumera todas as soluções possíveis, na **implícita** se enumera baseado na solução de subproblemas, diminuindo a quantidade de soluções possíveis.

Programação dinâmica consiste em armazenar soluções para subproblemas em uma tabela, para evitar que se calcule informações desnecessariamente. Isso pode diminuir o número de recursões, melhorando consideravelmente a complexidade de um algoritmo.

Algoritmo guloso é um algoritmo que procura sempre alcançar um resultado esperado a partir de uma heurística, mesmo que esse resultado não seja o melhor. Este tipo de algoritmo analisa apenas o próximo passo e toma uma decisão supondo a que estaria mais próxima do resultado final.

Backtracking é considerar outros caminhos em uma árvore de decisões, ou seja, voltar atrás em passos já tomados para tomar decisões diferentes.

Questão 9:

- a) O problema da satisfatibilidade consiste em atribuir valores a uma expressão booleana na forma normal conjuntiva afim de verificar se existe uma solução verdadeira.

$$(\neg x_1 \vee \neg x_2) \wedge (x_1 \vee \neg x_2) \\ x_1 = T, \quad x_2 = F$$

- b) As classes P, NP, NP-Completo e NP-Difícil caracterizam problemas quanto à sua solução em tempo polinomial.

P é solucionável em tempo polinomial.

NP não é solucionável em tempo polinomial.

NP-Difícil são problemas que não são solucionáveis em tempo polinomial.

NP-Completo é a interseção entre NP e NP-Difícil: representa problemas que têm solução, mas não em tempo polinomial.