**Universidade Federal de Roraima**

**Departamento de Ciência da Computação**

**Análise de Algoritmos**

**2a Lista**

**Aluno: Rodrigo dos Santos Tavares**

**Questão 2:**



Método mestre:

**Questão 3:**

1. Grafos são estruturas usadas para representar relações entre objetos, sendo representados por , em que é o conjunto de vértices, e é o conjunto de arestas.

Por exemplo, o Grafo , pode ser representado da seguinte forma:

1

3

4

5

2

1. **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 2 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

1. 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.

1

5

6

4

2

3

5

6

2

3

*Grafo de exemplo*

*Vizinhança do vértice 1*

1. Um **grafo planar** é um grafo que pode ser desenhado no plano sem que nenhuma de suas arestas se cruzem.
2. Um **grafo completo** é um grafo em que todas os vértices são adjacentes uns aos outros.

1

3

4

5

2

**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 e de vértices, todas as arestas do grafo conectam vértices do conjunto com o conjunto .

1. 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.

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

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

**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 .

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 .

Como alternativa à lista, pode-se usar também árvores balanceadas para os valores das chaves, garantindo uma menor complexidade .

**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:**

1. O problema da satisfatibilidade consiste em atribuir valores a uma expressão booleana na forma normal conjuntiva afim de verificar se existe uma solução verdadeira.
2. 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.