

Algoritmos e Estruturas de Dados

Caros alunos, as videoaulas desta disciplina encontram-se no AVA (Ambiente Virtual de Aprendizagem).



Unidade 4

Árvores e Grafos

Introdução da Unidade

Olá aluno, nesta unidade, continuaremos em constante evolução do conteúdo programático, aprofundando um pouco mais o desenvolvimento de algoritmos estruturados nesta disciplina.

Será apresentado nesta Unidade de Ensino o conceito de Árvores e o conceito de Grafos, importantes estruturas de dados utilizadas no desenvolvimento de algoritmos nos diversos cenários existentes.

A estrutura de uma árvore dentro de algoritmos, se assemelha bastante à uma árvore genealógica, que representa um histórico dos antepassados de uma pessoa. Portanto, também será semelhante a uma árvore convencional. Veremos isso no decorrer da Unidade.

Os conceitos dessas duas estruturas são utilizados em algoritmos para resolver problemas computacionais, entre eles, os algoritmos de busca.

O que vamos estudar realmente, é como implementar a ideia desses algoritmos para solucionar alguns problemas computacionais.

A nossa disciplina segue uma política de evolução dos conteúdos, logo, os conhecimentos prévios para esta unidade são os conteúdos que foram abordados nas Unidades 1, 2 e 3 dessa disciplina. Os conteúdos sobre *Arrays* Unidimensionais e os algoritmos de busca são fundamentais para o estudo das estruturas de Árvore e Grafos presentes na Unidade 4.

Este conteúdo, objetivando maximizar o processo de ensino e de aprendizagem, também será apresentado por meio de uma linguagem dialógica, facilitando assim a sua compreensão. Para esta unidade, a linguagem de programação Java continuará sendo utilizada no desenvolvimento do conteúdo e nos exercícios das videoaulas, mas sempre é bom lembrar, que os conteúdos estudados são aplicáveis a qualquer outra linguagem de programação, desde que, seja respeitada os particulares da linguagem em questão.

Objetivos

- Conhecer e compreender a estrutura de Árvore;
- Conhecer e compreender a estrutura de Grafos;
- Implementar algoritmos manipulando informações por meio da estrutura Árvore;
- Implementar algoritmos manipulando informações por meio da estrutura Grafos.

Conteúdo programático

Aula 01 – Algoritmos de Árvore e Árvore binária

Aula 02 – Algoritmos de Grafos



Você poderá também **assistir às videoaulas** em seu celular! Basta apontar a câmera para os **QRCodes** distribuídos neste conteúdo.

Pode ser necessário instalar um aplicativo de leitura QRcode no celular e efetuar login na sua conta Gmail.

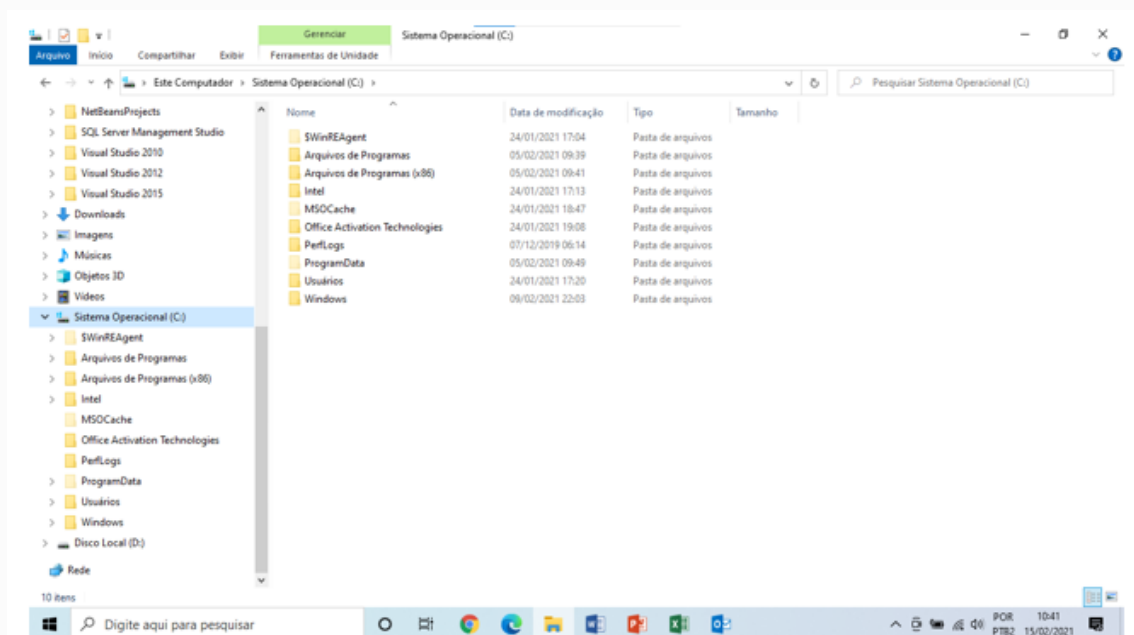
Aula 1 – Algoritmos de Árvore e Árvore binária

Introdução à estrutura de Árvore

Vamos começar nosso estudo nesta Unidade de Ensino, falando da estrutura de Árvore. Esta estrutura é uma das mais famosas no meio algorítmico e é aplicada quando precisamos representar dados que devem ser dispostos de maneira hierárquica em um sistema. Podemos citar como exemplo, uma hierarquia de diretórios ou uma Árvore genealógica.

Na área de sistemas operacionais (S.O.), a hierarquia de diretórios é muito comum. O Windows Explorer é uma interface gráfica do S.O. Windows, para exibição de diretórios e arquivos armazenados em um computador.

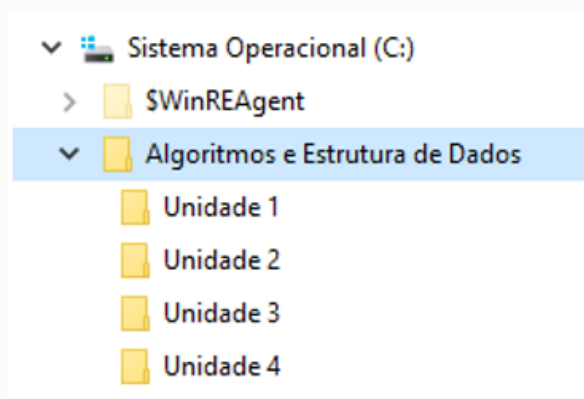
Figura 1 – Windows explorer



Fonte: elaborada pelo próprio autor (2020) S.O. - Windows 10.

A figura 1, traz um exemplo do Windows Explorer, no qual podemos observar que os diretórios estão inseridos dentro da "Unidade C:". Isso quer dizer que se quisermos criar diretórios ou inserir arquivos em diretórios já existentes, temos que respeitar a hierarquia presente. A unidade C do S.O., é também conhecida popularmente como "C Raiz", o mesmo nome no qual o nodo principal é conhecido dentro do conceito de Árvore. Isso significa que não existe nada antes dele.

Figura 2 – Hierarquia de diretórios



Fonte: elaborada pelo autor (2020) S.O. - Windows 10

Na figura 2, os diretórios Unidade 1, Unidade 2, Unidade 3 e Unidade 4 foram criados dentro do diretório Algoritmos e Estrutura de Dados, que por sua vez está dentro do diretório “C Raiz”. Isso reforça que a hierarquia dentro da árvore do Windows Explorer é respeitada.

Outro exemplo do cotidiano de representação de dados, que devem ser dispostos de maneira hierárquica, é a árvore genealógica. É a maneira mais clara de se explicar qual a origem da família de uma pessoa, ou seja, seus ancestrais. Vale lembrar que para criar sua árvore genealógica, devemos seguir alguns princípios básicos. Para explicar melhor, vamos citar e dar uma visão geral de cada um deles:

- **Nome e sobrenome:** vários registros podem ser pesquisados por nomes, no entanto, com o passar dos anos, as pessoas podem mudar de nome por vários motivos. Em alguns países, o Brasil por exemplo, quando a mulher se casa é opcional acrescentar o sobrenome do marido. Um outro exemplo, seria uma pessoa mudar o seu nome por causa da religião ou para esconder sua própria identidade. E não podemos descartar a possibilidade existente hoje, de alteração do nome, simplesmente pelo simples fato de não gostar dele;
- **Datas e locais de nascimento, casamento e registros de óbitos:** o nome da pessoa que você está buscando não vai ajudar muito se ele for muito comum, por exemplo José. É preciso uma ideia de onde e quando a pessoa morou ou viveu;
- **Formação acadêmica:** a educação de uma pessoa pode revelar muito sobre sua vida. Escolas e universidades mantêm um registro de seus alunos, sendo assim, pode-se encontrar informações importantes sobre seus familiares e construir sua árvore genealógica com mais precisão;
- **Ocupação:** é possível encontrar a casa em que um parente morou de acordo com informações de sua ocupação. Isso é possível pois, os censos, registros mantidos pelo governo de um país, podem rastrear a população por nomes, data de

nascimento, idade no último aniversário e residência atual, bem como a ocupação de uma pessoa;

- **Datas e locais de imigração:** se você é descendente de imigrantes, vale a pena descobrir quando e onde seus ancestrais vieram morar no local em que você nasceu. Os registros de imigração geralmente mencionam de onde o imigrante veio. Isso vai te ajudar a entender melhor a sua cultura e pode te dar pistas para que você encontre outra geração;
- **Características físicas:** a cor dos olhos, pele e cabelos, podem revelar um conjunto de características semelhantes de seus ancestrais. Essas características físicas também dizem muito quando você encontrar uma pessoa para continuar a construção de sua árvore genealógica.

Após seguir esses princípios você terá mais sucesso para construir sua árvore genealógica com mais fidelidade e número de ancestrais envolvidos. Vejamos o exemplo da figura 3.

Figura 3 – Árvore genealógica



Fonte: disponível em: <https://education.myheritage.com>. Acesso em: 12 dez. 2020.

A figura 3, exibe o esquema gráfico, de uma árvore genealógica. A figura, ainda, se assemelha bastante a uma abstração da estrutura de dado “Árvore” no cenário de algoritmos.

Estrutura de Árvore

A estrutura de árvore representa um conjunto finito de elementos em que cada elemento é chamado de nó e o primeiro elemento da árvore recebe o nome de raiz. Como já abordado, os elementos de uma árvore ficam organizados de forma hierárquica, onde existe um único elemento raiz que fica no topo da árvore. Todos os outros elementos ficam abaixo do elemento raiz e são subordinados a ele, denominados “nós filhos” (ASCENCIO, ARAUJO, 2010).

Como em uma família convencional, os “nós filhos” podem conter zero, um ou mais filhos. Todos os nós filhos que não contém outros nós filhos recebem o nome de folha ou nós folha (ASCENCIO, ARAUJO, 2010).

Dentro da sua logística de processamento, a estrutura de árvore utiliza ponteiros de memória para representar os nós filhos (ASCENCIO, ARAUJO, 2010).

Indicação de Vídeo

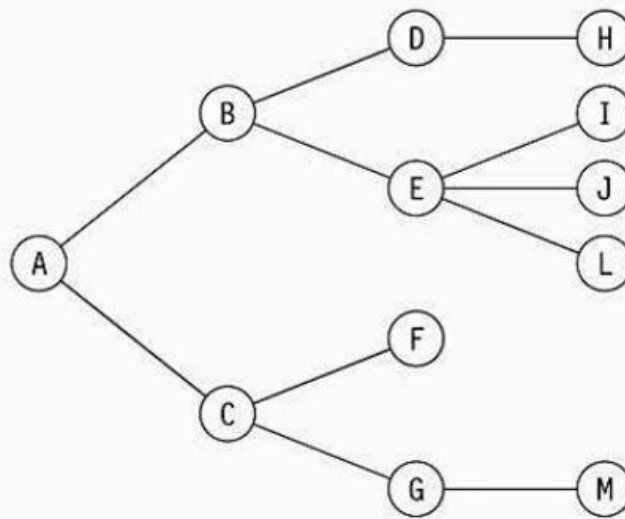
Veja este vídeo que traz uma explicação sobre a estrutura de Árvore. No vídeo, são abordadas as principais características desta estrutura algorítmica e alguns pontos de atenção sobre como ela funciona.

Vale a pena assistir este vídeo para que você saiba identificar e compreender a estrutura na prática.

Disponível em: <https://www.youtube.com/watch?v=PgZflufXGUU>. Acesso em: 30 dez. 2020.

Vamos apresentar na próxima imagem um exemplo de uma árvore e fazer algumas observações sobre a sua estrutura.

Figura 4 – Exemplo de árvore



Fonte: Forbellone (2005, p. 169).

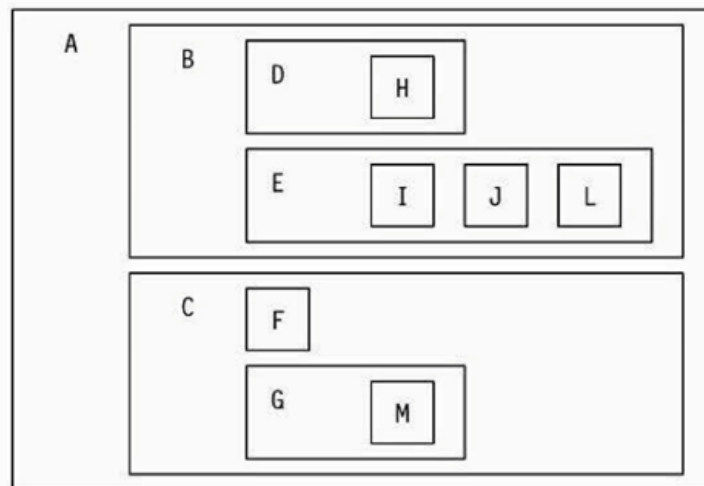
Na figura 4, podemos observar claramente que o topo da árvore, a Raiz, é o elemento A. Vamos apontar algumas características da árvore e adotar a nomenclatura correta para cada particularidade.

Características de uma árvore

- **Raiz:** toda a árvore possui o nó raiz, elemento que inicia a árvore, geralmente colocado no topo;
- **Grau:** o número máximo de ramificações a partir de um nó é chamado de Grau;
- **Nível ou Altura:** quantidade de níveis a partir do nó até o último nó é denominado altura ou profundidade. A diferença entre Nível e Grau é que o Nível é a partir da raiz e o Grau a partir de suas ramificações ou filhos.
- **Pai ou Ancestral:** o único antecessor de um nó é chamado de Pai;
- **Filhos ou Descendentes:** são os sucessores de um determinado nó, esses recebem o nome de Filhos;
- **Folhas:** são os nós que não tem descendentes, ou seja, sem filhos.

Sendo assim, podemos observar a figura 4, que D e E são descendentes de B, assim como D é pai de H. A árvore possui raiz em A, altura 4, grau 3 e os nós folhas são: F, H, I, J, L e M. Uma outra representação da mesma árvore pode ser observada na figura 5.

Figura 5 – Outra representação de árvores



Fonte: Forbellone (2005, p. 170).

Curiosidades – Árvore Binária

A árvore binária, assunto que abordaremos a seguir, contém algumas operações, são elas:

1. Inserir um nó na árvore;
2. Remover um nó de uma árvore;
3. Consultar os nós da árvore em ordem;
4. Consultar os nós da árvore em pré-ordem;
5. Consultar os nós da árvore em pós-ordem.

Na operação de inserção, a ordem dos elementos deve ser obedecida, assim como, todo nó inserido é uma folha.

Nas consultas, em ordem, “pré-ordem” e “pós-ordem”, são listados todos os nós, porém altera-se a ordem para cada tipo de consulta.

1- A consulta em ordem: a árvore é exibida com o ramo da esquerda, a raiz e o ramo da direita;

2- A consulta em pré-ordem: a árvore é exibida com a raiz, os ramos da esquerda e o ramo da direita;

3- A consulta em pós-ordem: a árvore é exibida com o ramo da esquerda, o ramo da direita e a raiz.

Fonte: disponível em: <http://www.dicio.com.br>.

Árvore Binária

A árvore binária é um conjunto finito de elementos divididos em 3 subconjuntos disjuntos. Raiz, sub árvore esquerda e sub árvore direita. Dentro de uma árvore binária, os nós só podem ter graus 0, 1 ou 2. Se for uma árvore binária completa todos os graus são iguais a 2 (ASCENCIO, ARAUJO, 2010).

Sendo uma árvore binária de busca, ela terá algumas regras pertinentes a ela. Na subárvore esquerda, ela terá elementos menores que a raiz, assim como, na subárvore direita, ela terá elementos maiores que a raiz. Isso vai facilitar muito quando quisermos implementar dentro desta árvore, a busca de um elemento específico (ASCENCIO, ARAUJO, 2010).

Dentro de uma árvore binária podemos realizar as seguintes operações: inserir um novo elemento, buscar um elemento, mostrar todos os elementos, remover um elemento, contar elementos ou calcular o nível de um elemento.

Vamos abordar um exemplo de um programa em Java que vai criar e implementar uma árvore binária.

Videoaula 1

Nesta videoaula, o professor vai apresentar a criação do projeto para implementar uma árvore binária. No vídeo, você verá a criação da classe “Nó” e seu construtor, ou seja, a criação do “Nó”. A construção do projeto, bem como, toda a explicação será por meio da linguagem de programação Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados.



Videoaula 1

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai apresentar a criação do projeto para implementar uma árvore binária.



Figura 6 – Classe No

```
5 public class No {  
6  
7     private int valor;  
8     private No esquerda;  
9     private No direita;  
10  
11     No(int valor, No esquerda, No direita){  
12  
13         this.valor = valor;  
14         this.esquerda = esquerda;  
15         this.direita = direita;  
16  
17     }  
18 }  
19  
20 }
```

Fonte: elaborada pelo próprio autor (2020).

A figura 6, demonstra o código da criação da classe “No” com a estrutura, e o construtor para criação de uma árvore. A próxima videoaula, vai se encarregar de criar o método para imprimir a árvore.

Videoaula 2

Nesta videoaula, o professor vai abordar a criação do método para impressão da árvore binária. Lembrando que a impressão da árvore, pode ser de 3 formas diferentes: em ordem, pré-ordem ou pós-ordem. No vídeo, você verá a explicação na sintaxe do Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados.

Este conteúdo, trabalha com métodos recursivos, portanto, é preciso ficar atento e acompanhar a explicação com muita atenção para compreender o assunto na sua totalidade.



Videoaula 2

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai abordar a criação do método para impressão da árvore binária. Lembrando que a impressão da árvore, pode ser de 3 formas diferentes: em ordem, pré-ordem ou pós-ordem.



Figura 7 – Método de impressão pré-ordem

```
18 public void imprimirPreOrdem(No n){
19
20     System.out.println(n.valor);
21
22     if (n.esquerda != null){
23         imprimirPreOrdem(n.esquerda);
24     }
25
26     if (n.direita != null){
27         imprimirPreOrdem(n.direita);
28     }
29
30 }
```

Fonte: elaborada pelo próprio autor (2020).

A figura 7, demonstra o código de impressão da árvore binária em pré-ordem. Na próxima videoaula, será exibido o código para criação da classe de teste para finalizarmos a aplicação.

Videoaula 3

Nesta videoaula, o professor vai abordar a criação da classe de teste que será responsável por controlar as árvores binárias. No vídeo, você contará com a explicação na sintaxe do Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados, além de dicas do professor para executar o código.

Esta videoaula é muito importante, pois fechará o ciclo entre o conceito e prática de uma árvore binária.



Videoaula 3

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai abordar a criação da classe de teste que será responsável por controlar as árvores binárias.



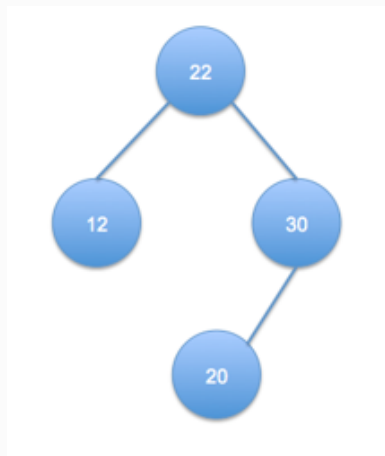
Figura 8 – Método de impressão pré-ordem

```
8 public static void main(String[] args) {  
9  
10     No no = new No(22, new No(12, null, null), new No(30, new No(20, null, null), null));  
11  
12     System.out.println("Pré-Ordem");  
13     no.imprimirPreOrdem(no);  
14  
15 }
```

Fonte: elaborada pelo próprio autor (2020).

Se considerarmos uma árvore binária criada a partir dos elementos da Figura 8, teremos a seguinte saída por parte do nosso programa: (22, 12, 30 e 20). Observe a figura 9 que apresenta a árvore do nosso exemplo.

Figura 9 – Esquema da árvore abordada



Fonte: elaborada pelo próprio autor (2020).

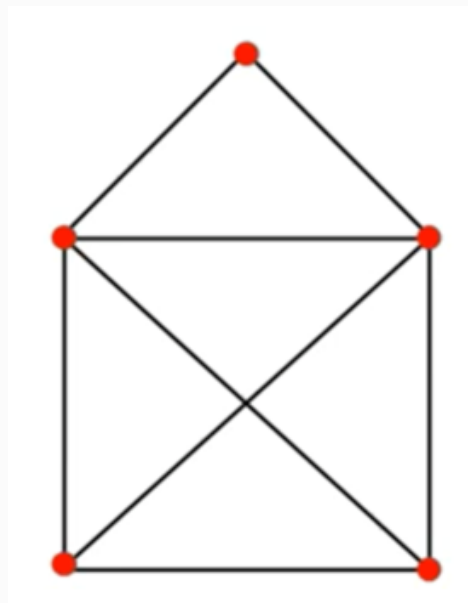
Aula 2 – Algoritmos de Grafos

Introdução à estrutura de Grafos

A estrutura de um grafo se parece muito visualmente com a estrutura de uma árvore, porém com características e funcionalidades diferentes. As árvores, como já abordado na aula 1 desta Unidade de Ensino, trabalha com a hierarquia dos elementos. Como exemplo, abordamos uma árvore genealógica e uma árvore de diretórios, como a encontrada hierarquicamente no *Windows Explorer*. O grafo por sua vez, mesmo que seus elementos também possam ser chamados de “Nós”, como na árvore, está mais relacionado com o caminho percorrido entre dois “Nós”, ou entre dois vértices.

Mas antes de explicar tecnicamente para que serve ou o que é um grafo, vamos considerar uma abordagem um pouco mais lúdica para explicar o que chamamos de “Teoria do Grafos”.

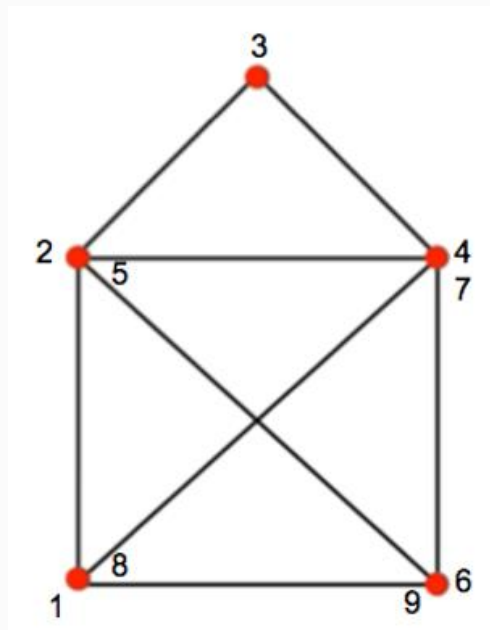
Figura 10 – Pontos Ligados



Fonte: elaborada pelo próprio autor (2020).

A ideia dos grafos, vem antes mesmo do seu significado. Em uma brincadeira de infância, a figura 10, apresenta uma lista de pontos ligados em que o objetivo era reproduzir o mesmo desenho sem tirar o lápis do papel e sem passar pelo mesmo caminho mais de uma vez. Sabendo como funciona a “Teoria dos Grafos”, é possível explicar se esse problema pode ser resolvido ou não. No caso da figura 10, a resposta é sim, desde que você comece o desenho por um ponto específico. Vamos exibir a mesma imagem agora sinalizando o caminho percorrido para que o problema possa ser resolvido.

Figura 11 – Pontos Ligados resolvido



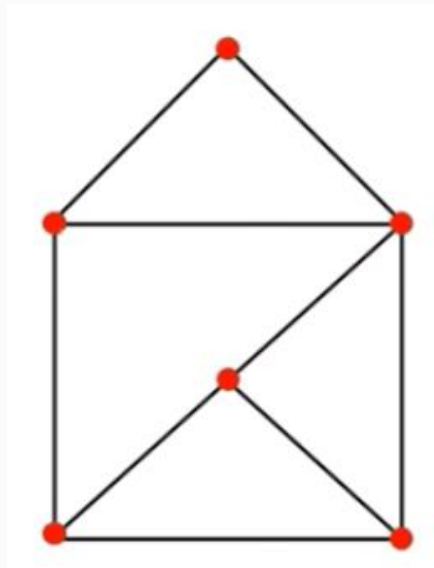
Fonte: elaborada pelo próprio autor (2020).

Na figura 11, basta você seguir a ordem crescente dos números, do 1 até o 9, para que você possa ver que é sim possível, sem tirar o lápis do papel e sem passar pelo mesmo caminho mais de uma vez, completar o problema.

Assim como, podemos dizer que é impossível completar o mesmo desafio com a imagem da Figura 12. Seguindo as mesmas regras, ou seja, sem tirar o lápis do papel e sem percorrer o mesmo caminho mais de uma vez.

Novamente quem me possibilita afirmar que este problema é impossível de resolver é a “Teoria dos Grafos”. Vamos a imagem.

Figura 12 – Pontos Ligados resolvido



Fonte: elaborada pelo próprio autor (2020).

Portanto, na Figura 12, não importa a ordem e nem o ponto onde começamos a resolver o problema, pois com as mesmas regras adotadas ele não tem solução.

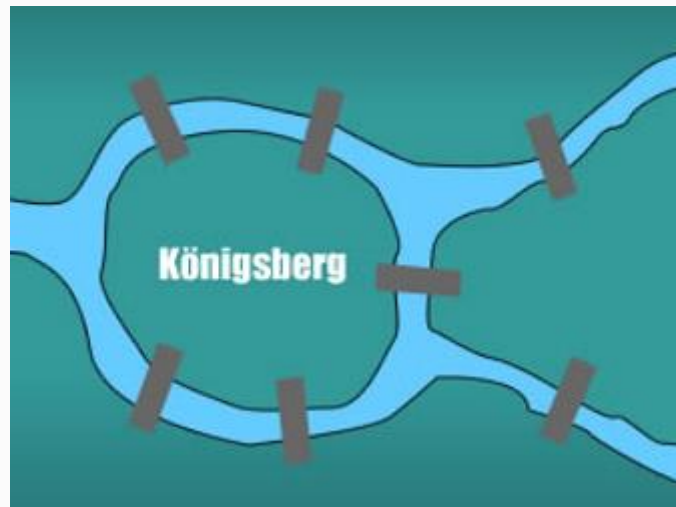
Teoria dos Grafos

Um dos problemas mais famosos sobre este tema é o das “Pontes de Königsberg” de quase 300 anos atrás, mais precisamente no ano de 1735. Königsberg, foi uma importante cidade da Prússia, localizada ao norte da Europa e próxima à costa do Mar Báltico e hoje tem outro nome, Kaliningrado.

Nessa cidade tinha 2 ilhas, margeadas pelo rio Pregel. A cidade também tinha 6 pontes que ligavam às margens e mais 1 que fazia a ligação das duas ilhas entre si. Contudo, por volta do ano de 1735, os moradores dessa cidade se perguntavam se existia como fazer um passeio percorrendo por todas as pontes, retornando ao ponto de partida, porém sem passar pela mesma ponte mais que uma vez.

A Figura 13, exibe um modelo contendo as 2 ilhas, todas as 7 pontes e os lados da cidade margeadas pelo rio.

Figura 13 – Esquema da cidade de Königsberg



Fonte: disponível em: <http://redeglobo.globo.com/globociencia/noticia/2011/12/entenda-o-enigma-das-pontes-de-konigsberg-que-instigou-geometria.html>. Acesso em: 30 dez. 2020.

Foi partindo disso, que o matemático Leonhard Euler, que viveu nessa época, se encarregou de tentar resolver esse problema.

Euler, utilizou a matemática para provar por “A + B” qualquer que fosse o resultado do problema, ou seja, se era ou não possível realizar o feito.

No detalhe, uma imagem de Leonhard Euler, matemático e físico suíço que falava alemão. Entre suas maiores descobertas, se encontra a teoria dos grafos.

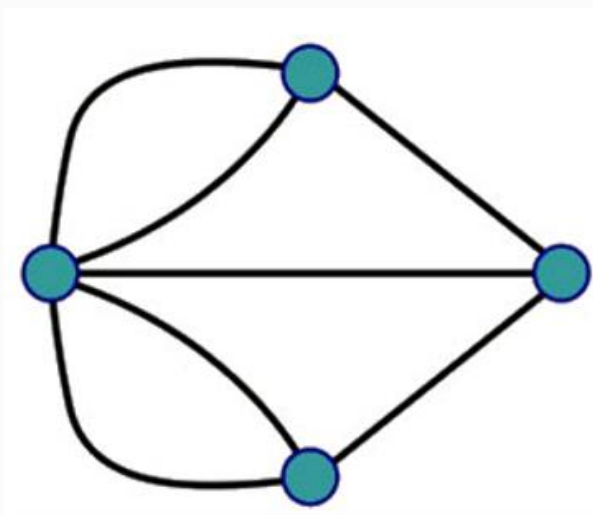
Figura 13 – Leonhard Euler



Fonte: disponível em: <https://www.bbc.com/portuguese/geral-44157282>. Acesso em: 30 dez. 2020.

Ele simplesmente eliminou os detalhes de distância física entre as pontes, paisagens, comprimento das pontes, o tamanho das ilhas etc.; portanto, Euler criou um esquema que recebeu o nome de Grafo. O problema das “Pontes de Königsberg”, foi modelado por ele utilizando o Grafo da Figura 14.

Figura 14 – Grafo de Leonhard Euler



Fonte: elaborada pelo próprio autor (2020).

Nesta aula, já associamos o grafo à estrutura de árvore, pois bem, como podem observar, a Figura 14 não deixa dúvidas disso. Cada ponto do grafo, pode ser comparado aos elementos da árvore, e cada linha do grafo também pode ser associado às linhas que ligam os nós filhos ao pai ou a raiz.

Voltando ao exemplo das pontes, o grafo de Euler tinha quatro pontos, dois deles representavam as duas margens do rio, e os outros dois, representavam as duas ilhas contidas na cidade. Diante disso, Euler modelou o gráfico com quatro pontos e sete linhas, que representavam as pontes, figura 14. Dentro deste modelo o desafio ainda era o mesmo, partir de um dos quatro pontos do grafo, percorrendo as sete linhas uma única vez e voltar ao ponto de partida.

A partir desse grafo, Euler provou que era impossível resolver tal problema, pois para atravessar qualquer ponto do grafo, são gastos duas pontes, uma para entrar no ponto do grafo e outra para sair. Contudo cada ponto do grafo, deve ter grau par de linhas ou de pontes. E no grafo representando o problema das “Pontes de Königsberg”, tem pontos de grau ímpar.

Indicação de Leitura

Para melhor compreensão deste tópico, você pode estudar o livro de (ASCENCIO, ARAUJO, 2010). O capítulo 8 “Algoritmos em Grafos”, especificamente, a partir da página 368, que apresenta a estrutura de Grafos abordando os conceitos, características e como resolver problemas de algoritmos com grafos. Vale a pena conferir.

ASCENCIO, A. F. G.; ARAÚJO, G. A. S. **Estruturas de dados**: algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo: Pearson Prentice Hall, 2010.

Após a explicação histórica da sua criação, como surgiu a ideia do grafo, quem o criou e como podemos nos beneficiar para resolvermos problemas complexos com grafos, vamos para uma abordagem um pouco mais prática, do ponto de vista algorítmico. Para tanto, devemos também aprender e compreender quais são as regras de criação, bem como a manipulação de elementos nessa estrutura.

Indicação de Vídeo

Veja este vídeo bem interessante sobre as principais características dos Grafos. No vídeo, você ficará por dentro das várias formas de representação de um grafo, suas classificações e características, grafos direcionados e não direcionados, grau, simetria entre outros conceitos.

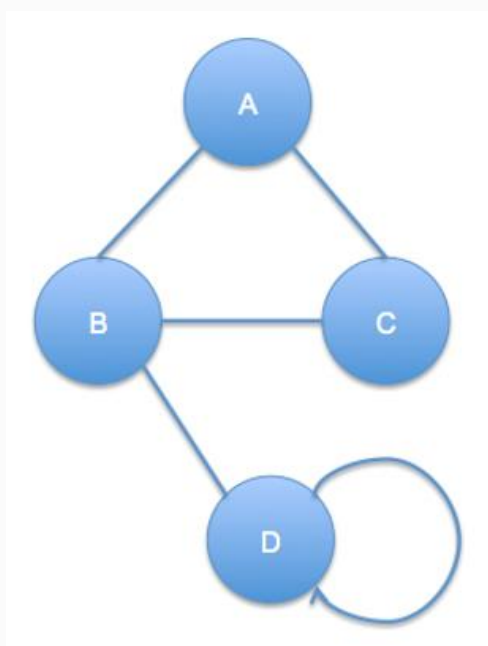
Por meio de um exemplo prático no cenário de grafos, o vídeo também traz um exemplo de programação com a linguagem Java, facilitando assim a compreensão da estrutura de dados. Pause o vídeo quantas vezes for necessário e faça anotações para que você compreenda o processo realizado.

Disponível em: <https://www.youtube.com/watch?v=Drd8rzVTUVo>. Acesso em: 20 dez. 2020.

Grafos

Um Grafo é uma estrutura de dados formada por um conjunto de vértices ou nós e um conjunto de arestas ligando aos vértices (ASCENCIO, ARAUJO, 2010).

Figura 15 – Exemplo de um Grafo



Fonte: elaborada pelo próprio autor (2020).

A Figura 15, apresenta no Grafo os vértices: A, B, C e D. Como pode ser observado, existem arestas, linhas que ligam os vértices entre: A e B, A e C, B e C e B e D. Também existe uma aresta circular ligando o vértice D nele mesmo. Como isso pode ser explicado na prática?

No cenário de grafos, além do exemplo das “Pontes de Königsberg”, a questão do tráfego urbano é utilizada para explicar um grafo, ou seja, para dar sentido na prática em todo este cenário. Imagine que você tenha que ir de uma estação do metrô “A” para outra estação “D”, mesmo que você tenha que passar por outras estações no decorrer do percurso, o seu destino final continua sendo a estação “D”. As estações são os vértices e o percurso de uma estação a outra são as arestas. Para tanto, a Figura 15 também explica este exemplo (ASCENCIO, ARAUJO, 2010).

Já o vértice “D”, que faz uma ligação com ele mesmo, podemos colocar o exemplo de um terminal urbano, no qual uma pessoa entra em um ônibus, passeia pela cidade e retorna até o ponto de saída. Neste caso, o ponto de saída e o ponto de destino são os mesmos. E esse exemplo é totalmente possível de acontecer, logo ele também deve ser modelado.

Ainda no Grafo da Figura 15, ressalto que ele também pode ser representado de forma matemática, dentro dos conjuntos “V” e “A”. Onde “V” representa o conjunto de vértices e “A” o conjunto de arestas (ASCENCIO, ARAUJO, 2010):

$$V = \{A, B, C, D\}$$

$$A = \{(A,B), (A,C), (B,C), (B,D), (D,D)\}$$

Os vértices são A, B, C e D, porém os relacionamentos ou arestas, representam as ligações entre os vértices. Vale lembrar que está sendo representada a ligação do vértice “D” com ele mesmo no final da representação matemática (ASCENCIO, ARAUJO, 2010).

Vamos abordar um programa na linguagem de programação Java demonstrando um exemplo de grafos na prática.

Videoaula 1

Nesta videoaula, o professor vai abordar a criação da classe Grafo. No vídeo, será abordada a criação da classe, a criação das listas de vértices e listas de arestas necessárias para a criação da classe de grafos. Também será abordado a criação do projeto e organização do código fonte dentro de pacotes. O professor vai explicar o código, sua funcionalidade e fará alguns testes modificando alguns parâmetros no programa. Toda a explicação será na sintaxe do Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados.

Vale a pena conferir mais essa videoaula!



Videoaula 1

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai abordar a criação da classe Grafo. No vídeo, será abordada a criação da classe, a criação das listas de vértices e listas de arestas necessárias para a criação da classe de grafos.



Dando continuidade ao nosso exemplo vamos abordar a criação das classes “Vertice” e “Aresta”.

Videoaula 2

Nesta videoaula, o professor vai abordar a criação das classes “Vertice” e “Aresta “. No vídeo, será abordado a criação dessas classes, bem como seus métodos e atributos. O professor vai explicar o código continuando o exemplo da videoaula anterior. Sendo assim, no final desta videoaula teremos as três classes criadas com seus métodos e aplicações pertinentes.

A explicação será na sintaxe do Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados.

Vamos continuar nosso exemplo agregando informações importantes em mais esta videoaula!



Videoaula 2

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai abordar a criação das classes “Vertice” e “Aresta “. No vídeo, será abordado a criação dessas classes, bem como seus métodos e atributos.



Depois de estudarmos a criação de todas as classes necessárias em nosso projeto, chegou a hora de testá-lo e ver se está tudo funcionando corretamente.

Videoaula 3

Nesta videoaula, o professor vai abordar a execução do código apresentado como exemplo para controle do exemplo de grafos. No vídeo, o professor vai explicar o código e realizar possíveis alterações de execução, visando apresentar outras maneiras de resolver o problema do algoritmo.

A explicação continua sendo na sintaxe do Java, Linguagem de Programação de apoio à disciplina de Algoritmos e Estrutura de Dados.

Vamos continuar nosso exemplo agregando informações importantes em mais esta videoaula!



Videoaula 3

Utilize o QRcode para assistir!

Nesta videoaula, o professor vai abordar a execução do código apresentado como exemplo para controle do exemplo de grafos.



Encerramento da Unidade

Chegamos ao fim de mais uma Unidade de Ensino após caminhar por conteúdos fundamentais para o seu conhecimento e desenvolvimento na disciplina de Algoritmos e Estrutura de Dados. A Unidade 4, apresentou uma das estruturas mais famosas e complexas dentro da disciplina de Algoritmos e Estrutura de Dados.

A estrutura de Árvore, e também a estrutura de Grafo, ambas apresentadas na prática dentro dos algoritmos, foi muito importante para agregar o seu aprendizado.

Durante essa Unidade, foram apresentados os processos de criação e manipulação de dados dentro dessas estruturas. A Unidade 4 trouxe ainda na prática, uma visão geral sobre os problemas cotidianos que podem ser resolvidos com essas estruturas.

Após a assimilação dos conteúdos abordados nesta Unidade, você estará apto a prosseguir estudando, pesquisando e aprofundando o seu conhecimento em Algoritmos e Estrutura de Dados.

Ótimos estudos, e não se esqueça de verificar e estudar a Bibliografia utilizada e as aulas gravadas pelo professor.

Obrigado pela oportunidade de ser o mediador entre os conteúdos específicos abordados e o conhecimento adquirido por você por meio da didática e abordagem desta Unidade.



Videoaula Encerramento

Utilize o QRcode para assistir!

Videoaula Encerramento



Referências

ASCENCIO, A. F. G.; ARAÚJO, G. A. S. **Estruturas de dados**: algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo: Pearson Prentice Hall, 2010.

CHEN, P. P. S. **The entity-relationship model—toward a unified view of data**. ACM Transactions on Database System, v. 1, p. 9–36, 1976.

DEITEL, Harvey M.; DEITEL, Paul J. **Java**: como programar. 8. ed. -. São Paulo: Pearson, 2010.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação**. 2. ed. São Paulo: Makron Books, 2000, 195 p.

LAUREANO, M. **Estrutura de dados com algoritmos e C**. São Paulo: Brasport, 2012.

MANZANO, J. A. N. G. **Estudo Dirigido**: algoritmos. São Paulo: Érica, 1997, 265 p.

Esperamos que este guia o tenha ajudado compreender a organização e o funcionamento de seu curso. Outras questões importantes relacionadas ao curso serão disponibilizadas pela coordenação.

Grande abraço e sucesso!

