

Algoritmos de Grafo

□ schmittjoaopedro . Java □ 27 27+00:00 Junho 27+00:00 20134 04+00:00 Novembro 04+00:00 2017 □ 1 Minute
Olá,

Grafos são sistemas de interconexões que trabalham com vértices que são os pontos de partida e chegada e arestas que são propriamente o caminho entre os vértices, podemos fazer uma analogia de comparação de grafos com o sistema neural humano.

Atualmente os grafos tem grande abrangência na área tecnológica pois muitos sistemas distribuídos fazem uso dos mesmos. Exemplo os GPS, Mapas e a própria rede de internet global trabalha em cima de grafos.

Em uma rede global de internet os grafos são representados da seguinte maneira, um roteador representa um vértice e qualidade do link de transporte entre dois roteadores é a aresta do grafo, assim, fazendo uma análise temos um grafo ponderado, que trabalha a ideia de calcular o melhor caminho até determinado roteador, neste exemplo de grafo entre roteadores os links apresentam latência e quantidade de banda que ponderam a distância entre dois roteadores, criando assim uma teia de caminhos que podemos usar para calcular rotas.

Abaixo segue projeto de aplicação feita no NetBeans IDE demonstrando a criação de um grafo e o cálculo de menor caminho, neste projeto são usados conceitos de busca em largura e o algoritmos dijkstra para calcular menor caminho.

O código fonte está disponível no seguinte [link \(https://github.com/schmittjoaopedro/algoritmos-grafo-java\)](https://github.com/schmittjoaopedro/algoritmos-grafo-java).

Segue abaixo implementação parcial algoritmos dijkstra pela Ana Fernanda Gomes:

```
static void dijkstra(listaadj Adj[], int tam, int v) {  
    int i, w;  
    int C[] = new int[tam];  
    int tamC = 0;  
    lista = new ListaPriori(tam);  
    dist[v] = 0;  
    lista.inserir(v, dist);  
    for (i = 1; i <= tam; i++) {  
        if (i != v) {  
            dist[i] = Integer.MAX_VALUE;  
            pai[i] = 0;  
            lista.inserir(i, dist);  
        }  
    }  
}
```

```

while (lista.tam != 0) {
w = lista.remover(dist);
C[tamC] = w;
tamC++;
    vertice x = Adj[w].listav;
while (x != null) {
relax(w, x.num, x.peso);
x = x.prox;
}
lista.constroiheap(dist);
}
}
static void relax(int u, int v, int peso) {
if (dist[v] > dist[u] + peso) {
dist[v] = dist[u] + peso;
pai[v] = u;
}
}
}

```

Autor: João Pedro Schmitt 27/06/2013

Com as etiquetas :

Algoritmos,
 Aplicacao,
 Aresta,
 Busca,
 Catolica,
 codigo,
 Computadores,
 dados,
 Dijkstra,
 estrutura,
 Google Maps,
 GPS,
 grafis,
 Grafo Ponderado,
 Grafos,
 java,
 Java SE,
 Largura,
 Mapas,
 Ponderado,
 programa,
 Redes,
 Roteadores,
 Vértice

Publicado por schmittjoaopedro



Graduado como bacharel em Sistemas de Informação pelo Centro Universitário Católica de Santa Catarina campus Jaraguá do Sul. Formado no Ensino Médio pelo Senai com Técnico em Redes de Computadores Articulado. Atualmente desenvolvedor JEE/Web em Sistemas de Engenharia na WEG. Pesquisador no período de faculdade em Informática pela Católica de Santa Catarina. Contato 47 - 99615 2305 E-mail: schmittjoaopedro@gmail.com Web page: <https://joaoschmitt.wordpress.com/> Linkedin: <https://www.linkedin.com/in/joao-pedro-schmitt-60847470/> Curriculum lattes: <http://lattes.cnpq.br/9304236291664423> Twitter: @JooPedroSchmitt [Ver todos os artigos de schmittjoaopedro](#)

