Estrutura de Dados (CCA410)

Aula 11 - Caminhos Mínimos (Dijkstra, Bellman-Ford)

Prof. Luciano Rossi
Prof. Leonardo Anjoletto Ferreira
Prof. Flavio Tonidandel
Prof. Fabio Suim

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2025

Grafos

Classificação das Estruturas de Dados

Primitivas	Compostas		
	Simples	Lineares	Não-Lineares
Inteiro	Cadeia (string)	Pilha	Árvore
Real	Registro (struct)	Fila	**Grafo**
Lógico	Arranjo (array)	Lista	
Caractere			

Caminhos Mínimos

Caminhos Mínimos

Definição

Problema dos Caminhos Mínimos

Dado um grafo ponderado $G = (V, E)$ com função de peso $w : E \to \mathbb{R}$, encontrar o caminho de menor custo entre dois vértices. O peso de um caminho é a soma dos pesos de suas arestas.

centro universitário FEI

Caminhos Mínimos

Aplicações

Aplicações Práticas

- Sistemas de navegação GPS
- Roteamento de redes de computadores
- Logística e transporte
- Jogos e simulações
- Análise de redes sociais

centro universitário FEI

Algoritmo de Dijkstra

Algoritmo de Dijkstra

Características

Algoritmo de Dijkstra

- **Vantagem:** Número menor de relaxamentos através de fila de prioridade
- **Restrição:** Funciona apenas com grafos com pesos não-negativos
- **Estratégia:** Algoritmo guloso que sempre escolhe o vértice mais próximo

centro universitário FEI

Algoritmo de Dijkstra

Atributos dos Vértices

Atributos

- $v.d$: Estimativa da distância mínima da origem até v
- $v.pai$: Predecessor de v na árvore de caminhos mínimos
- $v.cor$: Estado do vértice (BRANCO, CINZA, PRETO)

centro universitário FEI

Algoritmo em Pseudocódigo

centro universitário FEI

Algoritmo de Dijkstra

Pseudocódigo

INITIALIZE-SINGLE-SOURCE(G, s)

```
1 para cada vértice v ∈ G.V faça
2     v.d = ∞
3     v.pai = NIL
4 s.d = 0
```

RELAX(u, v, w)

```
1 se v.d > u.d + w(u, v)
2     então v.d = u.d + w(u, v)
3     v.pai = u
```

DIJKSTRA(G, w, s)

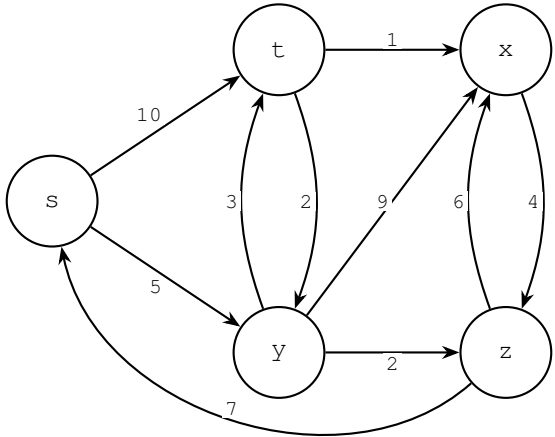
```
1 INITIALIZE-SINGLE-SOURCE(G, s)
2 S = ∅
3 Q = G.V
4 enquanto Q ≠ ∅ faça
5     u = EXTRACT-MIN(Q)
6     S = S ∪ {u}
7     para cada vértice v ∈ G.Adj[u] faça
8         RELAX(u, v, w)
```

centro universitário FEI

Simulação

Algoritmo de Dijkstra

Exemplo



Algoritmo de Dijkstra

Propriedades e Limitações

Propriedades

+ Sempre encontra solução ótima
+ Eficiente para grafos esparsos
+ Implementação relativamente simples
+ Constrói árvore de caminhos mínimos
+ Aplicável a grafos direcionados e não-direcionados

Limitações

- Não funciona com pesos negativos
- Complexidade alta para grafos densos
- Requer estrutura de fila de prioridade
- Não detecta ciclos negativos
- Calcula caminhos de um único vértice origem

centro universitário FEI

Quando Usar Dijkstra

Use Dijkstra quando tiver um **grafo com pesos não-negativos** e precisar dos caminhos mínimos a partir de um único vértice origem. É a escolha ideal para sistemas de navegação, roteamento de redes e logística.

Algoritmo de Dijkstra

Considerações finais

Estruturas de Dados

- **Fila de Prioridade:** Heap binário ou heap de Fibonacci
- **Representação do Grafo:** Lista de adjacência (recomendada)
- **Arrays de Distância:** Para armazenar $d[v]$
- **Array de Predecessores:** Para reconstruir caminhos

Otimizações

- **Heap de Fibonacci:** $O(V \log V + E)$
- **Parada Antecipada:** Quando destino é encontrado
- **Dijkstra Bidirecional:** Busca simultânea origem-destino
- **A*:** Com heurística para direcionamento

centro universitário FEI

Algoritmo de Bellman-Ford

centro universitário FEI

Algoritmo de Bellman-Ford

Características

Algoritmo de Bellman-Ford

- **Vantagem:** Funciona com grafos que possuem pesos negativos
- **Detecção:** Identifica a presença de ciclos negativos
- **Estratégia:** Relaxa todas as arestas $|V|-1$ vezes sistematicamente

Algoritmo de Bellman-Ford

Atributos dos Vértices

Atributos

- $v.d$: Estimativa da distância mínima da origem até v
- $v.pai$: Predecessor de v na árvore de caminhos mínimos
- **Inicialização:** Todos os vértices com $d = \infty$, exceto origem com $d = 0$

Algoritmo em Pseudocódigo

centro universitário FEI

Algoritmo de Bellman-Ford

Pseudocódigo

INITIALIZE-SINGLE-SOURCE (G, s)

```
1 para cada vértice v ∈ G.V faça
2       v.d = ∞
3       v.pai = NIL
4 s.d = 0
```

RELAX (u, v, w)

```
1 se v.d > u.d + w(u, v)
2       então v.d = u.d + w(u, v)
3       v.pai = u
```

BELLMAN-FORD (G, w, s)

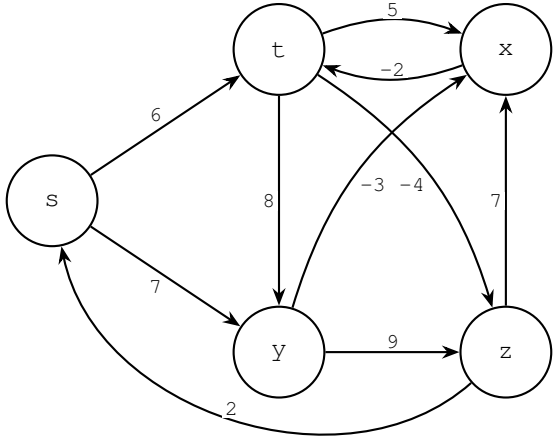
```
1 INITIALIZE-SINGLE-SOURCE(G, s)
2 para i = 1 até |G.V| − 1 faça
3       para cada aresta (u, v) ∈ G.E faça
4           RELAX(u, v, w)
5 para cada aresta (u, v) ∈ G.E faça
6       se v.d > u.d + w(u, v)
7           retorna FALSO
8 retorna VERDADEIRO
```

centro universitário FEI

Simulação

Algoritmo de Bellman-Ford

Exemplo



Algoritmo de Bellman-Ford

Propriedades e Limitações

Propriedades

+ Funciona com pesos negativos
+ Detecta ciclos negativos
+ Sempre encontra solução ótima (se existir)
+ Implementação mais simples que Dijkstra
+ Não requer estruturas auxiliares complexas

Limitações

- Complexidade alta: $O(V \cdot E)$
- Mais lento que Dijkstra para grafos sem pesos negativos
- Não funciona com ciclos negativos
- Pode ser ineficiente para grafos densos
- Sempre executa $|V| - 1$ iterações

centro universitário FEI

Algoritmo de Bellman-Ford

Considerações finais

Quando Usar Bellman-Ford

Use Bellman-Ford quando tiver um **grafo com possíveis pesos negativos** ou quando precisar detectar ciclos negativos. É essencial para aplicações financeiras, análise de arbitragem e sistemas com custos negativos.

centro universitário FEI

Algoritmo de Bellman-Ford

Considerações finais

Estruturas de Dados

- **Arrays de Distância:** Para armazenar $d[v]$
- **Array de Predecessores:** Para reconstruir caminhos
- **Lista de Arestas:** Representação simples do grafo
- **Sem Fila de Prioridade:** Diferente do Dijkstra

Comparação com Dijkstra

- **Pesos Negativos:** Bellman-Ford (sim), Dijkstra (não)
- **Ciclos Negativos:** Bellman-Ford detecta, Dijkstra não
- **Eficiência:** Dijkstra mais rápido para pesos não-negativos

centro universitário FEI

Estrutura de Dados (CCA410)

Aula 11 - Caminhos Mínimos (Dijkstra, Bellman-Ford)

Prof. Luciano Rossi
Prof. Leonardo Anjoletto Ferreira
Prof. Flavio Tonidandel
Prof. Fabio Suim

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2025