

Análise de Algoritmos

**Parte destes slides são adaptações de slides
do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.**

Grafos e BFS

CLRS 22 Elementary Graph Algorithms

CLRS 22.1 e 22.2

Grafos

Grafo: par (V, E)

onde V é um conjunto (finito) de **vértices**

e E é um conjunto de **arestas**, que são pares de vértices.

Grafos

Grafo: par (V, E)

onde V é um conjunto (finito) de **vértices**

e E é um conjunto de **arestas**, que são pares de vértices.

Se cada **aresta** é um par *ordenado* de **vértices**,
o grafo é **dirigido**, senão é **não dirigido**.

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

$|V|$ listas, uma para cada vértice v de G :

$\text{adj}[v]$: lista dos vértices que são adjacentes a v
(ou seja, vértices u tais que (u, v) ou $\{u, v\} \in E$)

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

$|V|$ listas, uma para cada vértice v de G :

$\text{adj}[v]$: lista dos vértices que são adjacentes a v
(ou seja, vértices u tais que (u, v) ou $\{u, v\} \in E$)

Se G é dirigido, então $|E|$ entradas, senão $2|E|$ entradas.

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

$|V|$ listas, uma para cada vértice v de G :

$\text{adj}[v]$: lista dos vértices que são adjacentes a v
(ou seja, vértices u tais que (u, v) ou $\{u, v\} \in E$)

Se G é dirigido, então $|E|$ entradas, senão $2|E|$ entradas.

Matriz de incidências:

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

$|V|$ listas, uma para cada vértice v de G :

$\text{adj}[v]$: lista dos vértices que são **adjacentes** a v
(ou seja, vértices u tais que (u, v) ou $\{u, v\} \in E$)

Se G é dirigido, então $|E|$ entradas, senão $2|E|$ entradas.

Matriz de incidências:

Matriz binária A de dimensão $|V| \times |V|$ onde

$A[u][v] = 1$ se e somente se (u, v) ou $\{u, v\} \in E$.

Representação de grafos

Grafo $G = (V, E)$.

Listas de adjacências:

$|V|$ listas, uma para cada vértice v de G :

$\text{adj}[v]$: lista dos vértices que são **adjacentes** a v
(ou seja, vértices u tais que (u, v) ou $\{u, v\} \in E$)

Se G é dirigido, então $|E|$ entradas, senão $2|E|$ entradas.

Matriz de incidências:

Matriz binária A de dimensão $|V| \times |V|$ onde

$A[u][v] = 1$ se e somente se (u, v) ou $\{u, v\} \in E$.

Grafos com **pesos** ou **comprimentos** nas arestas.

Busca em largura

Inicialização

BFS (G, s)

- 1 **para cada** $u \in V(G) \setminus \{s\}$ **faça**
- 2 $\text{color}[u] \leftarrow \text{branco}$
- 3 $d[u] \leftarrow \infty$
- 4 $\pi[u] \leftarrow \text{nil}$
- 5 $Q \leftarrow \emptyset$ $\triangleright$ fila dos vértices descobertos
- 6 $\text{color}[s] \leftarrow \text{cinzento}$
- 7 $d[s] \leftarrow 0$
- 8 $\pi[s] \leftarrow \text{nil}$
- 9 **INSIRA**(Q, s)

Busca em largura

BFS (G, s)

- 1 **para cada** $u \in V(G) \setminus \{s\}$ **faça**
- 2 $\text{color}[u] \leftarrow \text{branco}$ $d[u] \leftarrow \infty$ $\pi[u] \leftarrow \text{nil}$
- 3 $Q \leftarrow \emptyset$ $\triangleright$ fila dos vértices descobertos
- 4 $\text{color}[s] \leftarrow \text{cinzento}$ $d[s] \leftarrow 0$ $\pi[s] \leftarrow \text{nil}$
- 5 **INSIRA**(Q, s)

Busca em largura

BFS (G, s)

```
1  para cada  $u \in V(G) \setminus \{s\}$  faça
2       $\text{color}[u] \leftarrow \text{branco}$     $d[u] \leftarrow \infty$     $\pi[u] \leftarrow \text{nil}$ 
3   $Q \leftarrow \emptyset$             $\triangleright$  fila dos vértices descobertos
4   $\text{color}[s] \leftarrow \text{cinzento}$     $d[s] \leftarrow 0$     $\pi[s] \leftarrow \text{nil}$ 
5  INSIRA( $Q, s$ )

6  enquanto  $Q \neq \emptyset$  faça
7       $u \leftarrow \text{REMOVA}(Q)$ 
8      para cada  $v \in \text{adj}[u]$  faça
9          se  $\text{color}[v] = \text{branco}$ 
10             então  $\text{color}[v] \leftarrow \text{cinzento}$ 
11                  $d[v] \leftarrow d[u] + 1$ 
12                  $\pi[v] \leftarrow u$ 
13                 INSIRA( $Q, v$ )
14      $\text{color}[u] \leftarrow \text{preto}$ 
15 devolva  $d, \pi$ 
```

Descrição

Vértice branco: ainda não descoberto

Vértice cinzento: descoberto mas não processado
(são os vértices em Q)

Vértice preto: processado

Descrição

Vértice branco: ainda não descoberto

Vértice cinzento: descoberto mas não processado
(são os vértices em Q)

Vértice preto: processado

BFS devolve em π uma BF árvore enraizada em s .

$\pi[u]$: predecessor ou pai de u na BF árvore

Descrição

Vértice branco: ainda não descoberto

Vértice cinzento: descoberto mas não processado
(são os vértices em Q)

Vértice preto: processado

BFS devolve em π uma BF árvore enraizada em s .

$\pi[u]$: predecessor ou pai de u na BF árvore

$d[u]$: distância de s a u em G

BFS descobre todos os vértices à distância k
antes de descobrir qualquer um à distância $k + 1$

Consumo de tempo

Cada vértice é descoberto uma única vez, pois é branco e, ao ser descoberto, passa a ser cinzento, e depois preto.

Consumo de tempo

Cada vértice é descoberto uma única vez, pois é branco e, ao ser descoberto, passa a ser cinzento, e depois preto.

A lista de adjacência de cada vértice descoberto é percorrida uma única vez, quando o vértice sai de Q .

Consumo de tempo

Cada vértice é descoberto uma única vez, pois é branco e, ao ser descoberto, passa a ser cinzento, e depois preto.

A lista de adjacência de cada vértice descoberto é percorrida uma única vez, quando o vértice sai de Q .

Logo o consumo de tempo é $O(n + m)$, onde $n = |V|$ e $m = |E|$, pois a inicialização custa $\Theta(n)$ e a soma do tamanho das listas de adjacências percorridas é $O(m)$.

Consumo de tempo

Cada vértice é descoberto uma única vez, pois é branco e, ao ser descoberto, passa a ser cinzento, e depois preto.

A lista de adjacência de cada vértice descoberto é percorrida uma única vez, quando o vértice sai de Q .

Logo o consumo de tempo é $O(n + m)$, onde $n = |V|$ e $m = |E|$, pois a inicialização custa $\Theta(n)$ e a soma do tamanho das listas de adjacências percorridas é $O(m)$.

O consumo de tempo de uma BFS é linear no tamanho do grafo.

Correção

Vamos verificar que $d[u]$ é a distância de s a u e que a BF árvore contém um caminho mínimo de s a u .

Correção

Vamos verificar que $d[u]$ é a distância de s a u e que a BF árvore contém um caminho mínimo de s a u .

Seja $\delta(s, u)$ a distância de s a u em G .

Correção

Vamos verificar que $d[u]$ é a distância de s a u e que a BF árvore contém um caminho mínimo de s a u .

Seja $\delta(s, u)$ a distância de s a u em G .

Lema: $\delta(s, v) \leq \delta(s, u) + 1$ para toda aresta uv .

Correção

Vamos verificar que $d[u]$ é a distância de s a u e que a BF árvore contém um caminho mínimo de s a u .

Seja $\delta(s, u)$ a distância de s a u em G .

Lema: $\delta(s, v) \leq \delta(s, u) + 1$ para toda aresta uv .

Lema: $d[v] \geq \delta(s, v)$ para todo vértice ao final da BFS.

Correção

Vamos verificar que $d[u]$ é a distância de s a u e que a BF árvore contém um caminho mínimo de s a u .

Seja $\delta(s, u)$ a distância de s a u em G .

Lema: $\delta(s, v) \leq \delta(s, u) + 1$ para toda aresta uv .

Lema: $d[v] \geq \delta(s, v)$ para todo vértice ao final da BFS.

Lema: Se Q contém os vértices $v_1, v_2, \dots, v_r$ nesta ordem, então $d[v_r] \leq d[v_1] + 1$ e $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \dots, r - 1$.

Correção

Seja $\delta(s, u)$ a distância de s a u em G .

Lema: $\delta(s, v) \leq \delta(s, u) + 1$ para toda aresta uv .

Lema: $d[v] \geq \delta(s, v)$ para todo vértice ao final da BFS.

Lema: Se Q contém os vértices $v_1, v_2, \dots, v_r$ nesta ordem, então $d[v_r] \leq d[v_1] + 1$ e $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \dots, r - 1$.

Corolário: Se v_i entra na fila antes de v_j , então $d[v_i] \leq d[v_j]$ quando v_j entra na fila.

Correção

Seja $\delta(s, u)$ a distância de s a u em G .

Lema: $\delta(s, v) \leq \delta(s, u) + 1$ para toda aresta uv .

Lema: $d[v] \geq \delta(s, v)$ para todo vértice ao final da BFS.

Lema: Se Q contém os vértices $v_1, v_2, \dots, v_r$ nesta ordem, então $d[v_r] \leq d[v_1] + 1$ e $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \dots, r - 1$.

Corolário: Se v_i entra na fila antes de v_j , então $d[v_i] \leq d[v_j]$ quando v_j entra na fila.

Teorema: Ao final da BFS, $d[v] = \delta(s, v)$ para todo $v \in V$.
Ademais, para todo v alcançável de s com $v \neq s$,
um caminho mais curto de s a v consiste em
um caminho mais curto de s a $\pi[v]$ seguido da aresta $\pi[v]v$.