

Digrafos

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria de Grafos
Bacharelado em Ciência da Computação

01 de agosto de 2016

Plano de Aula

- 1 Pensamento
- 2 Revisão
- 3 Digrafos - Matriz de Adjacências
- 4 Digrafos - Lista de Adjacências

Sumário

- 1 **Pensamento**
- 2 Revisão
- 3 Digrafos - Matriz de Adjacências
- 4 Digrafos - Lista de Adjacências

Pensamento



Pensamento



Frase

As normas existem para a obediência dos tolos e para a orientação dos sábios.

Quem?

David Ogilvy (1911-1999)
Publicitário inglês.

Sumário

- 1 Pensamento
- 2 Revisão
- 3 Digrafos - Matriz de Adjacências
- 4 Digrafos - Lista de Adjacências

Conceitos

- Digrafo

Conceitos

- Digrafo
- Arco

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo:

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$
- Arcos paralelos e antiparalelos

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$
- Arcos paralelos e antiparalelos
- Leque de saída e de entrada

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$
- Arcos paralelos e antiparalelos
- Leque de saída e de entrada
- Grau de saída e de entrada

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$
- Arcos paralelos e antiparalelos
- Leque de saída e de entrada
- Grau de saída e de entrada
- Fonte, sorvedouro e vértice isolado

Conceitos

- Digrafo
- Arco
- Ponta (inicial e final)
- v-w (sair e entrar)
- Vizinhança
- Tamanho do digrafo: $V + A$
- Arcos paralelos e antiparalelos
- Leque de saída e de entrada
- Grau de saída e de entrada
- Fonte, sorvedouro e vértice isolado
- $A(G) \subset V(G)^2$

Conceitos

- Digrafo completo

Conceitos

- Digrafo completo
- Torneio

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador
 - Subdigrafo induzido

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador
 - Subdigrafo induzido
 - Subdigrafo próprio

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador
 - Subdigrafo induzido
 - Subdigrafo próprio
 - Subdigrafo induzido por X

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador
 - Subdigrafo induzido
 - Subdigrafo próprio
 - Subdigrafo induzido por X
- Superdigrafo

Conceitos

- Digrafo completo
- Torneio
 - em um torneio $\rightarrow A = V(V - 1)/2$
- Digrafo denso
- Digrafo esparso
- Digrafo simétrico = Grafo
- Subdigrafo
 - Subdigrafo gerador
 - Subdigrafo induzido
 - Subdigrafo próprio
 - Subdigrafo induzido por X
- Superdigrafo
- Um subdigrafo de um grafo não é necessariamente um grafo

Conceitos

- Representação dos vértices

Conceitos

- Representação dos vértices
- Representação dos arcos

Conceitos

- Representação dos vértices
- Representação dos arcos
 - Matriz de adjacências

Conceitos

- Representação dos vértices
- Representação dos arcos
 - Matriz de adjacências
 - Lista de adjacências

Sumário

- 1 Pensamento
- 2 Revisão
- 3 Digrafos - Matriz de Adjacências
- 4 Digrafos - Lista de Adjacências

Representação com Matriz de Adjacências

Estrutura básica

```
struct digraph {  
    int V;  
    int A;  
    int **adj;  
};
```

Representação com Matriz de Adjacências

Estrutura básica

```
struct digraph {  
    int V;  
    int A;  
    int **adj;  
};
```

Digrafo

```
typedef struct digraph *Digraph;
```

Representação com Matriz de Adjacências

Inicialização

```
Digraph DIGRAPHinit( int V ) {  
    Digraph G = malloc( sizeof *G);  
    G->V = V;  
    G->A = 0;  
    G->adj = MATRIXint( V, V, 0);  
    return G;  
}
```


Representação com Matriz de Adjacências

Inicialização da matriz

```
static int **MATRIXint( int r, int c, int val) {  
    Vertex i, j;  
    int **m = malloc( r * sizeof (int *));  
    for (i = 0; i < r; i++)  
        m[i] = malloc( c * sizeof (int));  
    for (i = 0; i < r; i++)  
        for (j = 0; j < c; j++)  
            m[i][j] = val;  
    return m;  
}
```

Representação com Matriz de Adjacências

Inserção

```
void DIGRAPHinsertA( Digraph G, Vertex v, Vertex w) {  
    if (G->adj[v][w] == 0) {  
        G->adj[v][w] = 1;  
        G->A++;  
    }  
}
```

Representação com Matriz de Adjacências

Inserção

```
void DIGRAPHinsertA( Digraph G, Vertex v, Vertex w) {  
    if (G->adj[v][w] == 0) {  
        G->adj[v][w] = 1;  
        G->A++;  
    }  
}
```

Remoção

```
void DIGRAPHremoveA( Digraph G, Vertex v, Vertex w) {  
    if (G->adj[v][w] == 1) {  
        G->adj[v][w] = 0;  
        G->A--;  
    }  
}
```

Representação com Matriz de Adjacências

Exibição

```
void DIGRAPHshow( Digraph G) {  
    Vertex v, w;  
    for (v = 0; v < G->V; v++) {  
        printf( "%2d:", v);  
        for (w = 0; w < G->V; w++)  
            if (G->adj[v][w] == 1)  
                printf( " %2d", w);  
        printf( "\n");  
    }  
}
```

Sumário

- 1 Pensamento
- 2 Revisão
- 3 Digrafos - Matriz de Adjacências
- 4 Digrafos - Lista de Adjacências

Representação com Lista de Adjacências

Estrutura básica

```
struct digraph {  
    int V;  
    int A;  
    link *adj;  
};
```

Representação com Lista de Adjacências

Estrutura básica

```
struct digraph {  
    int V;  
    int A;  
    link *adj;  
};
```

Digrafo

```
typedef struct digraph *Digraph;
```

Representação com Lista de Adjacências

Nó e Ligação

```
typedef struct node *link;  
struct node {  
    Vertex w;  
    link next;  
};
```


Representação com Lista de Adjacências

Nó e Ligação

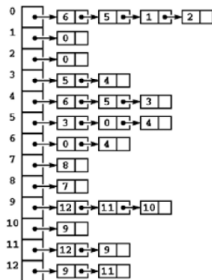
```
typedef struct node *link;  
struct node {  
    Vertex w;  
    link next;  
};
```

Novo nó

```
static link NEWnode( Vertex w, link next) {  
    link a = malloc( sizeof (struct node));  
    a->w = w;  
    a->next = next;  
    return a;  
}
```

Representação com Lista de Adjacências

Lista de adjacências



Representação com Lista de Adjacências

Inicialização

```
Digraph DIGRAPHinit( int V) {  
    Vertex v;  
    Digraph G = malloc( sizeof *G);  
    G->V = V;  
    G->A = 0;  
    G->adj = malloc( V * sizeof (link));  
    for (v = 0; v < V; v++)  
        G->adj[v] = NULL;  
    return G;  
}
```



Representação com Lista de Adjacências

Inserção

```
void DIGRAPHinsertA( Digraph G, Vertex v, Vertex w) {  
    link a;  
    for (a = G->adj[v]; a != NULL; a = a->next)  
        if (a->w == w) return;  
    G->adj[v] = NEWnode( w, G->adj[v]);  
    G->A++;  
}
```

Digrafos

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria de Grafos
Bacharelado em Ciência da Computação

01 de agosto de 2016