

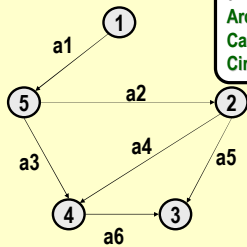
Grafos – aula 2

Estruturas de Dados - Grafos

Caminhamentos

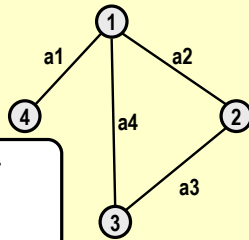
Estruturas de Dados - Grafos

Grafo orientado



$a = (v1, v2)$
Arco
Caminho
Circuito

Grafo não-orientado



$a = \{v1, v2\}$
Aresta
Cadeia
Ciclo

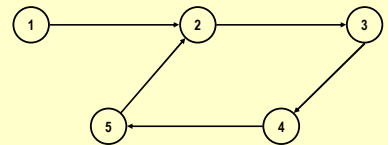
Estruturas de Dados - Grafos

Algoritmos

Problemas principais:

- Complexidade
- Ciclos

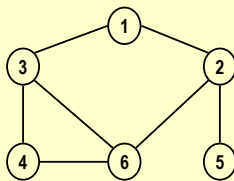
Ao contrário das listas lineares e das árvores, que são estruturas acíclicas, os grafos podem possuir **ciclos**



Estruturas de Dados - Grafos

Exame sistemático de vértices

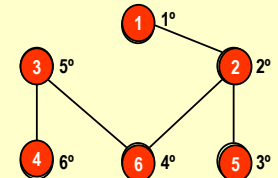
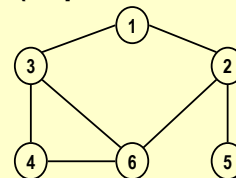
- Dado um grafo qualquer, como fazer para percorrer todos os seus vértices?
- Por onde começar?



Estruturas de Dados - Grafos

Exame sistemático de vértices

Percurso em profundidade (Depth First Search)



Estruturas de Dados - Grafos

Exame sistemático de vértices

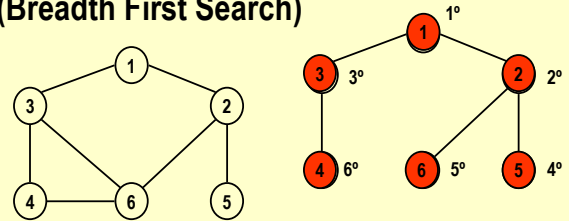
```

Procedimento DFS (G: PGrafo; v: PVertice)
{ Exame em Profundidade (Depth First Search) }
{ visitado: atributo do nodo }
início
  v↑.visitado ← 1; { nodo v escolhido arbitrariamente }
  para cada w adjacente a v
    faça se w↑.visitado = 0
      então DFS(G,w);
fim; { DFS }
    
```

Estruturas de Dados - Grafos

Exame sistemático de vértices

Percurso em amplitude (Breadth First Search)



Estruturas de Dados - Grafos

Exame sistemático de vértices

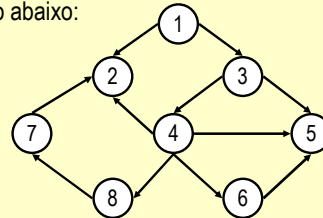
```

Procedimento BFS (G: PGrafo; v: PVertice)
{ Exame em Amplitude (Breadth First Search) }
{ visitado: atributo }
{ FV: Fila de vértices inicialmente vazia }
início
  v↑.visitado ← 1;
  insere(FV↑,v);
  enquanto FV não vazia
    faça início
      v ← retira(FV↑); { retira primeiro elemento da fila }
      para cada w adjacente a v
        faça se w↑.visitado = 0
          então início
            insere(FV↑,w);
            w↑.visitado ← 1;
          fim
    fim
fim; { DFS }
    
```

Estruturas de Dados - Grafos

Exercício

- Escreva a ordem dos vértices percorridos pelos caminhamentos em amplitude e profundidade para o grafo abaixo:



<http://www.cs.duke.edu/csed/jawaa2/examples/DFS.html>

<http://www.cs.duke.edu/csed/jawaa2/examples/BFS.html>

Estruturas de Dados - Grafos

Problema

- Desejamos conectar todos os computadores em um prédio de escritório usando a menor quantidade possível de cabos
- Como resolver o problema?

Estruturas de Dados - Grafos

Árvore geradora de grafo

Um **grafo** não-orientado G é uma **árvore** se satisfizer a qualquer uma das seguintes condições:

G é conexo e sem ciclos;
G é conexo, com n vértices e n-1 arestas.

Uma árvore obtida pela remoção dos ciclos de um grafo é chamada **árvore geradora do grafo**.

Estruturas de Dados - Grafos

Algoritmo de Paton

Objetivo: determinação de uma árvore geradora para um grafo não-valorado

A – arestas de G
V – vértices de G
AA – arestas da árvore geradora
VA – vértices da árvore geradora
X vértices ainda não examinados

Estruturas de Dados - Grafos

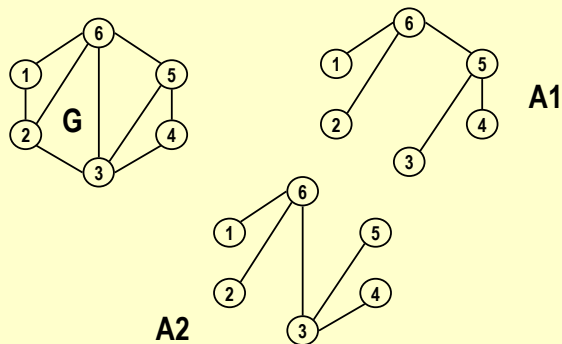
Algoritmo de Paton

Objetivo: determinação de uma árvore geradora para um grafo não-valorado

AA = 0; VA = 0; X = V;
1. Escolhe um vértice de V para raiz da árvore
2. Visita os vértices adjacentes a raiz
3. Repete-se mesmo procedimento para o último vértice visitado

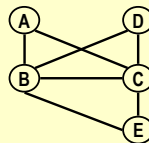
Estruturas de Dados - Grafos

Árvore geradora de grafo



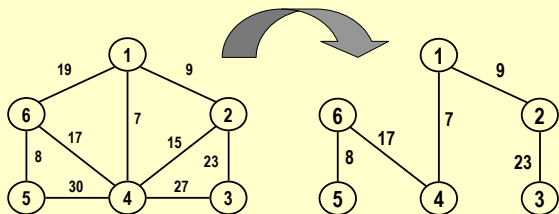
Estruturas de Dados - Grafos

Algoritmo de Paton



Estruturas de Dados - Grafos

Árvore geradora mínima



Árvore geradora mínima é, dentre as árvores geradoras, aquela que apresenta a menor soma dos valores associados às linhas.

Estruturas de Dados - Grafos

Algoritmo de Kruskal

Objetivo: determinação de uma árvore geradora para um grafo valorado

1. Iniciar com os n vértices e nenhuma aresta
2. Fila ordenada das arestas
3. Acrescentar uma aresta ainda não colocada e que não fecha ciclo
4. Repetir o passo 3 até que n-1 arestas tenham sido colocadas

–<http://students.ceid.upatras.gr/~papagel/project/kruskal.htm>

Estruturas de Dados - Grafos

Algoritmo de Kruskal

Geração de **árvore geradora mínima**:
escolhendo sempre os menores
valores de arcos disponíveis

O algoritmo exige que se tenha um método para determinar se uma aresta fecha ciclo ou não.

Estruturas de Dados - Grafos

Algoritmo de Kruskal

Algoritmo Kruskal (G);

G: Grafo não-orientado, sem laços, com n vértices $v_1 \dots v_n$ e m arestas $a_i = \{u_i, w_i\} \dots a_m = \{u_m, w_m\}$

A: Árvore de cobertura mínima

para cada vértice v em G faça

Defina um grupo elementar $C(v) := \{v\}$ { um grupo é uma lista não ordenada de vértices }

Inicialize uma fila ordenada Q para conter todas as arestas em G , usando seus pesos como chaves

Inicializa A

enquanto A tem menos de $n-1$ arestas faça

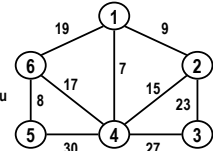
RetiraFila(u, v);

Seja $C(v)$ o grupo contendo v e $C(u)$ o grupo contendo u
se $C(v) \cap C(u) \neq \emptyset$ então

Coloque aresta (v, u) em A

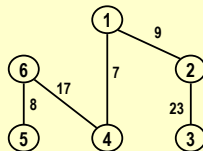
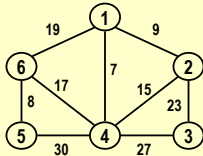
Unifique $C(v)$ e $C(u)$ em um grupo

Retorna a árvore A



Estruturas de Dados - Grafos

Algoritmo de Kruskal



Estruturas de Dados - Grafos

Exercício

- Existem 8 pequenas ilhas em um arquipélago e o governo deseja construir 7 pontes conectando-as de forma que cada ilha possa ser alcançada de qualquer outra ilha através de uma ou mais pontes
- O custo de construção de uma ponte é proporcional ao seu comprimento
- As distâncias entre os pares de ilhas são dados na tabela abaixo
- Ache quais pontes devem ser construídas para que o custo da construção seja mínimo

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	120
2	-	-	265	175	215	180	185	155
3	-	-	-	260	115	350	435	195
4	-	-	-	-	160	330	295	230
5	-	-	-	-	-	360	400	170
6	-	-	-	-	-	-	175	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

Estruturas de Dados - Grafos