

Aula 6

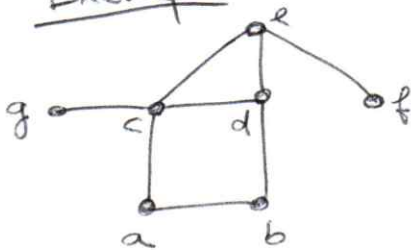
- Antes de iniciarmos o Cap. 3, vejamos alguns pontos restantes do Cap. 2

ÁRVORES (Seção 2.3)

Seja $G(V, E)$ um grafo.

- Denomina-se excentricidade de um vértice $u \in V$ ao valor máximo da distância entre u e w , $\forall w \in V$.

Exemplo



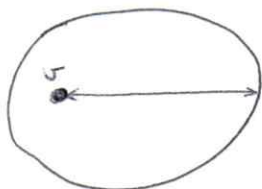
vértice	excentricidade
a	3
b	3
c	2
d	2
e	2
f	3
g	3

Exemplo



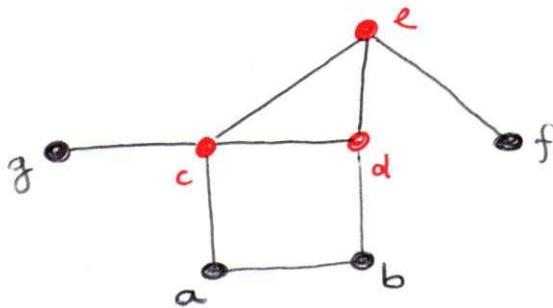
vértice	excentricidade
a	4
b	3
c	2
d	3
e	4

Analogia (da aula passada, com Prof. Fabiano)



- Denomina-se centro de G o subconjunto dos vértices com excentricidade mínima.

Nos exemplos anteriores:

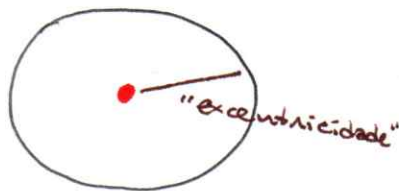


$\{c, d, e\}$ excentricidade 2



$\{c\}$ excentricidade 2

Analogia usada na aula do Prof. Fabiano:



- O centro de um grafo arbitrário pode ter de 1 a n vértices.
- O centro de uma árvore possui 1 ou 2 vértices.
(Teorema 2.5)

Prova Se a árvore tem até dois vértices, resultado trivial.
Caso contrário, aplica-se repetidamente o Lema abaixo.

Lema 2.1 Seja T um árvore com 3 ou mais vértices.

Seja T' a árvore obtida pela exclusão de todas as suas folhas. Então T e T' possuem o mesmo centro.

Prova (resumo):

- folha não pode pertencer ao centro
- dado um vértice interior v , o vértice de distância máxima w é folha
- remover todas as folhas diminui a excentricidade de todos os vértices interiores em 1 unidade.

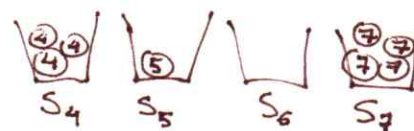
CAPÍTULO 3 - Técnicas Básicas

- Seção 3.2: discute processo de representação, etapa fundamental no início de algoritmos em grafos. É importante representar o grafo de forma adequada. Revejam a aula anterior.
- Seção 3.3: com frequência precisamos examinar a lista de adjacências $A(u)$, para $u \in V$, e perguntar se $w \in A(u)$ satisfaz uma certa propriedade.
- Seção 3.4: Ordenação em tempo linear
 - Lower bound do problema de ordenação por comparações é $\Omega(n \log n)$, em que n é o tamanho da sequência. Porém, se não forem feitas comparações entre os elementos é possível ordenar em tempo $O(n)$.
 - Com frequência precisamos ordenar vértices ou arestas de G segundo um certo critério.
 - Podemos usar ordenação por caixas.

Sejam S : conjunto de inteiros a serem ordenados

a : elemento mínimo de S

b : elemento máximo de S



Defina $S_a, S_{a+1}, \dots, S_b$: subconjuntos de S , inicialmente vazios

Algoritmo muito simples: para cada elemento de S , insira valor j na caixa S_j . Depois retire os elementos das caixas, começando por S_a .

Qual a complexidade da ordenação dos vértices segundo seus graus?

Coloração Aproximada

- Determinar exatamente o número cromático de um grafo é um problema muito difícil!
- Mas e se estivermos satisfeitos com uma coloração aproximada? Ou seja, com número de cores próximo do número cromático.

Tentativa:

Algoritmo 3.2: Coloração aproximada

dados: $G(V, E)$

Ordenar V em ordem não crescente de graus

Inicializar $C_1 := C_2 := \dots := C_n := \emptyset$

Colorir v_1 com cor 1, e incluir v_1 em C_1

Para $j = 2, 3, \dots, n$ efetuar *repete $O(n)$ vezes*

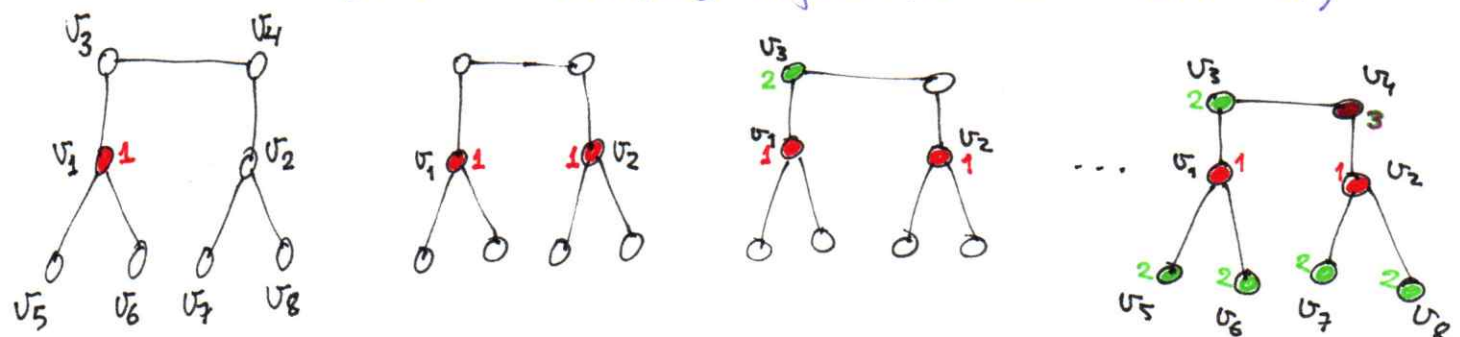
$r := \min \{ i \mid A(v_j) \cap C_i = \emptyset \}$ *consome $O(n)$ passos*

Colorir v_j com cor r , e incluir v_j em C_r

$O(n^2)$ operações

- O algoritmo acima é correto (retorna uma coloração)

Prova: Em qualquer passo j garantimos que os vértices $v_1, v_2, \dots, v_j$ estão coloridos corretamente (não há vértices adjacentes com mesma cor)



Outra tentativa:

Algoritmo 3.3: Coloração aproximada

dados: $G(V, E)$

$O(|A(v_j)|) +$
 $O(n)$

ordenar V em ordem não crescente de graus

Inicializar $f(1) := f(2) := \dots := f(n) := 0$

cada
passo:
 $O(|A(v_j)|)$
operações

para $j = 1, 2, \dots, n$ efetuar $\rightarrow$ repete para todo vértice v_j

para $w \in A(j)$ efetuar

se w já colorido então

$f(\text{con}(w)) := j$

ajusta o
vetor f

$f(k) = j$ diz que
 v_j já é adjacente
a vértice com
con k

$r := 1$

enquanto v_j não colorido efetuar

se $f(r) \neq j$ então

$\text{con}(v_j) := r$

caso contrário

$r := r + 1$

colore o
vértice
 v_j com
a "menor"
con ainda
"liberada"

complexidade $O(n + \sum |A(v_j)|) = O(n + m)$

- Não produz boas aproximações sempre

-Retomando Seção 2.8:

- Um conjunto parcialmente ordenado ou ordenação parcial, denotado $(S, \leq)$ é um conjunto não-vazio S e uma relação binária $\leq$ em S satisfazendo:

(i) $\Lambda_1 \leq \Lambda_1$ para $\Lambda_1 \in S$ (reflexivo)

(ii) $\Lambda_1 \leq \Lambda_2$ e $\Lambda_2 \leq \Lambda_1 \Rightarrow \Lambda_1 = \Lambda_2$ (anti-simétrico)

(iii) $\Lambda_1 \leq \Lambda_2$ e $\Lambda_2 \leq \Lambda_3 \Rightarrow \Lambda_1 \leq \Lambda_3$ (transitivo)

- POSET

- Alternativamente, podemos também definir um conjunto parcialmente ordenado pelo par $(S, <)$ em que

(i) $\Lambda_1 \not< \Lambda_1$ (irreflexivo)

(ii) $\Lambda_1 < \Lambda_2 \Rightarrow \Lambda_2 \not< \Lambda_1$ (assimétrico)

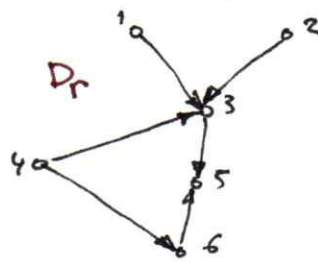
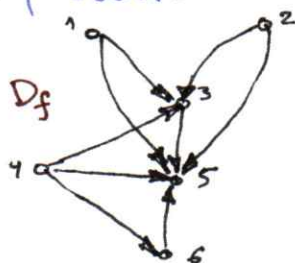
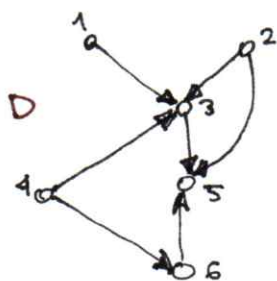
(iii) $\Lambda_1 < \Lambda_2$ e $\Lambda_2 < \Lambda_3 \Rightarrow \Lambda_1 < \Lambda_3$ (transitivo)

Por que isso é importante em nosso curso?
Por causa de digrafos acíclicos (DAG, na sigla em inglês)

Seja $D(V, E)$ um DAG

fecho transitivo é o maior digrafo $D_f(V, E_f)$ que preserva a alcançabilidade de D

redução transitiva é o menor digrafo $D_r(V, E_r)$ que preserva a alcançabilidade de D



Fecho transitivo de
Digrafo acíclico equivalente a Conjunto parcialmente ordenado

- Isto é, $P(S, <)$ caracteriza um Poset se e somente se o digrafo $D_f(S, <)$ for o fecho transitivo de algum digrafo acíclico D .
 - Diz-se que D induz o conjunto parcialmente ordenado $P(S, <)$.
 - Temos $\lambda_1 < \lambda_2$ em P sse existe caminho de λ_1 para λ_2 em D .
- Voltando ao capítulo 3:

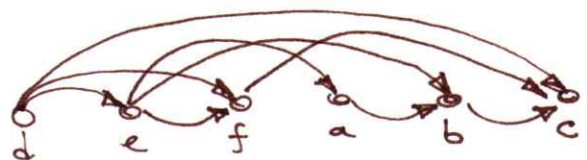
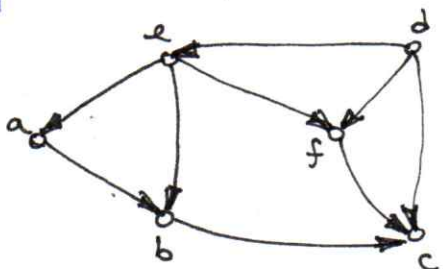
ORDENAÇÃO TOPOLÓGICA

- É possível ordenar os vértices de um DAG de modo a obter uma sequência $v_1, \dots, v_n$ tal que

$$v_i < v_j \Rightarrow i < j, \text{ para } 1 \leq i, j \leq n$$

Tal sequência é uma ordenação topológica

- Forma de desenhar todos os vértices de um DAG em linha reta com todas as arestas apontando para a frente



Algoritmo 3.4 : Ordenação topológica

dados: digrafo acíclico $D(V, E)$

para $j = 1, \dots, n$ efetuar

escolher um vértice w com grau de entrada nulo em D
retirar w de D junto com arestas divergentes
definir $v_j := w$

- Importante: todo DAG tem pelo menos um vértice com grau de entrada nulo!