CI065 — CI755
Algoritmos e Teoria dos Grafos

Alguns Exercícios Resolvidos

2 de agosto de 2016

2 1. Se G é um grafo de 14 vértices e 25 arestas cujos vértices tem graus 3 ou 5, quantos vértices tem grau 3 e quantos tem grau 5?

2. Generalize o raciocínio para um grafo de n vértices e m arestas cujos vértices tem graus d_1 ou d_2.

Resposta:

1.

$$\begin{aligned} 3t + 5c &= 2 \times 25, \\ t + c &= 14, \end{aligned}$$

logo

$$\begin{aligned} t &= 10, \\ c &= 4. \end{aligned}$$

2. Fazendo

n_1: # vértices de grau d_1, e

n_2: # vértices de grau d_2,

temos

$$\begin{aligned} n_1 d_1 + n_2 d_2 &= 2m, \\ n_1 + n_2 &= n, \end{aligned}$$

logo

$$n_1 = n - n_2,$$

e

$$(n - n_2)d_1 + n_2 d_2 = 2m,$$

ou seja

$$nd_1 + (d_2 - d_1)n_2 = 2m$$

e portanto,

$$\begin{aligned} n_2 &= \frac{2m - nd_1}{d_2 - d_1}, \\ n_1 &= \frac{nd_2 - 2m}{d_2 - d_1}. \end{aligned}$$

13 Prove que, se G é um grafo direcionado, então

$$\sum_{v \in V(G)} \delta^+(v) = |A(G)| = \sum_{v \in V(G)} \delta^-(v),$$

Resposta:

Seja G um grafo direcionado e seja M sua matriz de incidência. Vamos calcular a soma das entradas de M.

$$S(G) = \sum_{a \in A(G), v \in V(G)} M[v, a].$$

$$M : \begin{array}{c|cccc}
 & a_1 & a_2 & \dots & a_m \\
\hline
v_1 & M[v_1, a_1] & M[v_1, a_2] & \dots & M[v_1, a_m] \\
v_2 & M[v_2, a_1] & M[v_2, a_2] & \dots & M[v_2, a_m] \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
v_n & M[v_n, a_1] & M[v_n, a_2] & \dots & M[v_n, a_m]
\end{array}$$

Somando pelas linhas de M temos

$$S(G) = \sum_{v \in V(G)} \sum_{a \in A(G)} M[v, a],$$

e como

$$\sum_{a \in A(G)} M[v, a] = \delta_G^+(v) - \delta_G^-(v),$$

para cada $v \in V(G)$, então

$$S(G) = \sum_{v \in V(G)} \left(\delta_G^+(v) - \delta_G^-(v) \right). = \sum_{v \in V(G)} \delta_G^+(v) - \sum_{v \in V(G)} \delta_G^-(v).$$

Somando pelas colunas de M temos

$$S(G) = \sum_{a \in A(G)} \sum_{v \in V(G)} M[v, a],$$

e como

$$\sum_{v \in V(G)} M[v, a] = 0$$

para cada $a \in A(G)$, então

$$S(G) = \sum_{a \in A(G)} 0 = 0.$$

Então,

$$\sum_{v \in V(G)} \delta_G^+(v) - \sum_{v \in V(G)} \delta_G^-(v) = 0$$

ou seja,

$$\sum_{v \in V(G)} \delta_G^+(v) = \sum_{v \in V(G)} \delta_G^-(v).$$

Calculando do mesmo modo

$$S'(G) = \sum_{a \in A(G), v \in V(G)} |M[v, a]|,$$

chegamos a

$$\sum_{v \in V(G)} \delta_G^+(v) + \sum_{v \in V(G)} \delta_G^-(v) = 2|A(G)|,$$

e daí

$$\sum_{v \in V(G)} \delta_G^+(v) = 2|A(G)| = \sum_{v \in V(G)} \delta_G^-(v).$$

14 Prove que

$$M_{G^T} = (M_G)^T,$$

para todo grafo direcionado G, onde M^T denota a matriz transposta da matriz M.

Resposta:

Basta provar que

$$M_{G^T}[u,v] = (M_G)^T[u,v], \text{ para todo } u, v \in V(G).$$

Sejam $u, v \in V(G)$. Então

$$M_{G^T}[u,v] = 1$$

se e somente se

$$(u,v) \in A(G^T),$$

ou seja

$$(v,u) \in A(G),$$

e portanto

$$M_G[v,u] = 1,$$

ou seja

$$(M_G)^T[u,v] = 1.$$

30 Prove que um grafo G é bipartido se e somente se $E(G) = \partial_G(X)$ para algum $X \subseteq V(G)$ e que, neste caso, $\{X, V(G) - X\}$ é uma bipartição de G.

Resposta:

($\Rightarrow$) Considere uma bipartição X, Y de G. Vamos provar que $\partial_G(X) = E(G)$. Como é evidente que $\partial_G(X) \subseteq E(G)$, basta provar que $E(G) \subseteq \partial_G(X)$. Seja então $\{x, y\} \in E(G)$. Como G é bipartido, sem perda de generalidade podemos supor $x \in X$ e $y \in Y$ e portanto, $\{x, y\} \in \partial_G(X)$.

($\Leftarrow$) Seja $X \subseteq V(G)$ tal que $E(G) = \partial_G(X)$. Para provar quye G é bipartido, basta verificar que $X, V(G) - X$ é bipartição de G.

Como $X, V(G) - X$ é uma partição de $V(G)$, resta somente verificar que X e $V(G) - X$ são independentes.

Suponha que houvesse aresta $\{u, v\} \in E(G)$ com $u, v \in X$. Então $\{u, v\} \notin \partial_G(X)$ e contrariando a hipótese de que $E(G) = \partial_G(X)$.

Pelo mesmo argumento concluímos que $V(G) - X$ é independente.

78 Seja T a arborescência resultante de uma busca em largura sobre um grafo conexo G e seja r sua raiz. Prove que G é bipartido se e somente se $d(r,u)$ e $d(r,v)$ tem paridades diferentes para toda aresta $\{u,v\} \in G - E(S(T))$.

Resposta:

Seja T a arborescência resultante de uma busca em largura sobre um grafo conexo G e seja r sua raiz e seja $\{u,v\} \in G - E(S(T))$.

Como T é arborescência resultante de uma busca em largura, então (Corolário **??**) rTu e rTv são caminhos mínimos em G.

Como $rTu \cdot (u,v)$ é caminho de r a v em G, concluímos que

$$|rTv| \leq |rTu \cdot (u,v)| = |rTu| + 1.$$

Por argumento análogo, concluímos que

$$|rTu| \leq |rTv| + 1,$$

ou seja, os tamanhos dos caminhos rTu e rTv diferem de no máximo 1.

($\Rightarrow$) Se G é bipartido, então o circuito fundamental de $\{u,v\}$ com relação a T tem tamanho par e isso só é possível se os tamanhos dos caminhos rTu e rTv diferem de exatamente 1 e, portanto, $d(r,u)$ e $d(r,v)$ tem paridades diferentes.

($\Leftarrow$) Se $d(r,u)$ e $d(r,v)$ tem paridades diferentes para toda aresta $\{u,v\} \in G - E(S(T))$, então os conjuntos de vértices de G a distância par e ímpar de r são independentes e, portanto, G é bipartido.

84 Prove que se T é a arborescência resultante de uma busca em largura sobre um grafo conexo G e $r = r(T)$, então

$$|d_G(r,u) - d_G(r,v)| \leq 1, \text{ para todo } \{u,v\} \in E(G - S(T)).$$

Resposta:

Seja T a arborescência resultante de uma busca em largura sobre um grafo conexo G e seja $r = r(T)$. Seja ainda $\{u,v\} \in G - E(S(T))$.

Como T é arborescência resultante de uma busca em largura, então (Corolário **??**) rTu e rTv são caminhos mínimos em G e, consequentemente,

$$\begin{aligned} d_G(r,u) &= |rTu|, \text{ e} \\ d_G(r,v) &= |rTv|. \end{aligned}$$

Como $rTu \cdot (u,v)$ é caminho de r a v em G, concluímos que

$$|rTv| \leq |rTu \cdot (u,v)| = |rTu| + 1,$$

ou seja,

$$|rTv| - |rTu| \leq 1,$$

isto é,

$$d_G(r,v) - d_G(r,u) \leq 1,$$

Por argumento análogo, concluímos que

$$d_G(r,u) - d_G(r,v) \leq 1,$$

ou seja,

$$|d_G(r,u) - d_G(r,v)| \leq 1.$$

86 Seja T uma arborescência geradora de um grafo G produzida por uma busca em profundidade. Prove que

1. A raiz de T é vértice de corte de G se e somente se tem mais de um filho.
2. Um vértice v que não é raiz de T é vértice de corte de G se e somente se tem um filho w tal que nenhum descendente de w é vizinho de um ancestral próprio de v.
3. Uma aresta $\{u, v\}$ é aresta de corte em G se e somente se $(u, v) \in A(T)$ e nenhuma aresta em G liga um descendente de v a um ancestral de u.

Resposta:

Seja T uma arborescência geradora de um grafo G produzida por uma busca em profundidade e seja r sua raiz.

1. ($\Rightarrow$) Suponha que r é vértice de corte de G. Como r é raiz de T, então cada componente de $G - r$ tem que ter um filho de r. Se r é vértice de corte em G, então $G - r$ tem mais de um componente e, consequentemente, r tem mais de um filho em T.

   ($\Leftarrow$) Suponha que r é vértice de corte de G e tem u e v como filhos em T. Sejam U e V os descendentes de u e v em T, respectivamente. Os conjuntos U e V tem que ser disjuntos pois não pode haver vértice em T que seja descendente de ambos.
   Se houvesse caminho de u a v em $G - r$, teríamos uma aresta em $\partial_G(U)$ ligando um descendente de u a um não–descendente de u, ou seja, uma aresta cruzada em $G - E(S(T))$ com relação a T, o que não é possível pois T uma arborescência produzida por uma busca em profundidade.

2. ($\Rightarrow$) Seja v um é vértice de corte de G que não é raiz de T e seja p o pai de v. Nestas condições, o vértice v tem que ter um filho w em T tal que p e w estão em componentes diferentes de $G - v$.
   Suponha agora que w tem um descendente t que é vizinho de um ancestral próprio a de v. Neste caso $wTt \cdot (t, a) \cdot aTp$ seria caminho de w a p em $G - v$ e v não seria vértice de corte.

   ($\Leftarrow$) Seja v um vértice de T que tem um filho w tal que nenhum descendente de w é vizinho de um ancestral próprio de v. Se houvesse caminho de w ao pai de v em $G - v$, este caminho teria que ter ao menos uma aresta fora de $S(T)$. Como T é arborescência produzida por uma busca em profundidade esta aresta teria que ligar um descendente de w a um ancestral próprio de v.

3. ($\Rightarrow$) Seja $\{u, v\}$ uma aresta de corte em G. Sem perda de generalidade, podemos supor que u é o pai de v em T e, sendo assim, u é vértice de corte em G. De acordo com o item anterior, nenhuma aresta em G liga um descendente de v a um ancestral de u.

   ($\Leftarrow$) Seja $(u, v) \in A(T)$ tal que nenhuma aresta em G liga um descendente de v a um ancestral de u. Se houvesse caminho de u a v em $G - \{u, v\}$ este caminho teria uma aresta fora de $S(T)$. Como T é arborescência produzida por uma busca em profundidade, esta aresta teria que ligar um descendente de v a um ancestral u.

92 Seja G um grafo direcionado e considere uma execução do algoritmo $\mathsf{BuscaProfundidade}(G)$. Seja F a floresta direcionada induzida pelos valores de $v.\mathsf{pai} \mid v \in V(G)$ e, para cada $v \in V(G)$ sejam $v.\mathsf{pre}$ e $v.\mathsf{pos}$ os índices de pré–ordem e pós–ordem computados.

Prove que

1. Se (u, v) é arco de F ou arco de avanço com relação a F, então $u.\mathsf{pre} < v.\mathsf{pre} < v.\mathsf{pos} < u.\mathsf{pos}$.
2. Se (u, v) é arco cruzado com relação a F, então $v.\mathsf{pre} < v.\mathsf{pos} < u.\mathsf{pre} < u.\mathsf{pos}$.
3. A ordem $<$ induzida sobre $V(G)$ dada por

   $$u < v := u.\mathsf{pre} < v.\mathsf{pre}, \text{ para todo } u, v \in V(G),$$

   é uma pré–ordem de F.
4. A ordem $<$ induzida sobre $V(G)$ dada por

   $$u < v := u.\mathsf{pos} < v.\mathsf{pos}, \text{ para todo } u, v \in V(G),$$

   é uma pós–ordem de F.

Resposta:

1. Neste caso u é ancestral de v e, consequentemente $u.\mathsf{pre} < v.\mathsf{pre} < v.\mathsf{pos} < u.\mathsf{pos}$.
2. Neste caso a arborescência a que pertence o vértice v foi (toda ela) processada antes da arborescência de u e, consequentemente, $v.\mathsf{pre} < v.\mathsf{pos} < u.\mathsf{pre} < u.\mathsf{pos}$.
3. Sejam u e v tais que u é ancestral de v em F. Neste caso, temos que $u.\mathsf{pre} < v.\mathsf{pre}$ e, consequentemente, $u < v$. Assim, se u é ancestral de v, temos $u < v$ e, portanto, a ordem $<$ é uma pré–ordem de F.
4. Sejam u e v tais que u é ancestral de v em F. Neste caso, temos que $v.\mathsf{pos} < u.\mathsf{pos}$ e, consequentemente, $v < u$. Assim, se u é ancestral de v, temos $v < u$ e, portanto, a ordem $<$ é uma pós–ordem de F.

93 Prove que um grafo direcionado G admite ordenação topológica se e somente se é acíclico.

Resposta:

($\Rightarrow$) Seja G um grafo direcionado e seja $(v_1, \ldots, v_n)$ uma ordenação topológica de G. Se G não fosse acíclico, teria um circuito direcionado que poderia ser escrito como $(u_1, \ldots, u_k)$ de tal maneira que u_1 fosse o primeiro vértice de G segundo a ordenação topológica. Neste caso teríamos $(u_k, u_1) \in A(G)$ contrariando a ordenação topológica.

($\Leftarrow$) Vamos provar, por indução em $|V(G)|$, que se G é um grafo direcionado acíclico, então G admite ordenação topológica.

Base: Se $V(G) = \{v\}$, então (v) é uma ordenação topológica de G.

H.I.: Seja $a \in \mathbb{N}$ tal que todo grafo direcionado acíclico com até a vértices admite ordenação topológica.

Passo: Vamos provar que todo grafo direcionado acíclico com $a+1$ vértices admite ordenação topológica.

Seja G um grafo direcionado acíclico com $a + 1$ vértices.

Como G é acíclico, então G tem um vértice v que é fonte e $G - v$ é um grafo direcionado acíclico com a vértices.

Pela hipótese de indução, $G - v$ admite uma ordenação topológica $(v_1, \ldots, v_a)$ e daí, $(v, v_1, \ldots, v_a)$ será uma ordenação topológica de G já que v só tem vizinhos de saída em G.

95 O seguinte algoritmo que recebe um grafo direcionado G e devolve (uma lista com) a ordenação topológica de G ou um subgrafo de G sem fontes, caso G seja cíclico.

Qual o seu tempo de execução no pior caso (em termos assintóticos) em função de $|V(G)|$ e $|E(G)|$?

Com relação ao desempenho de pior caso (em termos assintóticos) como ele se compara ao algoritmo discutido em aula?

```
Ordena(G)

  Se V(G) = ∅
      Devolva ()
  Se G não tem fonte
      Devolva G
  v ← fonte de G
  R ← Ordena(G − v)
  Se R é uma lista
      acrescente v ao início de R
  Devolva R
```

```
Ordena(G)

  Q ← lista vazia
  Enquanto V(G) ≠ ∅
      Se G não tem fonte
          Devolva G
      v ← fonte de G
      remova v de G e enfile em Q
  Devolva Q
```

Resposta:

Considere uma implementação do Algoritmo Ordena() onde $V(G)$ seja uma fila de prioridades ordenada pelo grau dos vértices.

Haverá um rearranjo na fila de prioridades a cada aresta removida de G.

Cada rearranjo destes é feito em tempo $O(\log|V(G)|)$ e o número de remoções de arestas é no máximo $|E(G)|$.

Logo, o tempo de execução no pior caso é $O(|E(G)|\log|V(G)|)$.

Numa análise de pior caso, este algoritmo é mais lento que o algoritmo discutido em aula, que executa em tempo $O(|E(G)| + |V(G)|)$ no pior caso.

99 Seja M um emparelhamento em um grafo G e seja P um caminho M–aumentante. Prove que

1. O conjunto $M \oplus E(P)$ é um emparelhamento em G.
2. $|M \oplus E(P)| = |M| + 1$.

Resposta:

Seja M um emparelhamento em um grafo G e seja P um caminho M–aumentante em G.

1. Considere duas arestas em $M \oplus E(P) = (M - E(P)) \cup (E(P) - M)$. Considere as três possibilidades.

   ambas estão em $M - E(P)$: neste caso elas não tem vértices em comum, pois M é emparelhamento.

   ambas estão em $E(P) - M$: neste caso elas também não tem vértices em comum, pois P é caminho alternante.

   uma delas está em $M - E(P)$ e a outra em $E(P) - M$: neste caso elas não podem ter vértices em comum pois os vértices de uma estão em $V(P)$ e os da outra não estão.

2.

$$
\begin{aligned}
|M \oplus E(P)| &= |(M \cup E(P)) - (M \cap E(P))| \\
&= |M \cup E(P)| - |M \cap E(P)| \\
&= |M| + |E(P)| - 2|M \cap E(P)|.
\end{aligned}
$$

Como P é caminho M–aumentante, temos

$$|M \cap E(P)| = \frac{|E(P)| - 1}{2},$$

e daí,

$$
\begin{aligned}
|M \oplus E(P)| &= |M| + |E(P)| - 2|M \cap E(P)| \\
&= |M| + |E(P)| - 2\frac{|E(P)| - 1}{2} \\
&= |M| + |E(P)| - (|E(P)| - 1) \\
&= |M| + 1.
\end{aligned}
$$