1^{a} prova de Algoritmos e Teoria dos Grafos
2° semestre de 2005

---

(1) LEIA COM ATENÇÃO TODA A PROVA. Entender o enunciado das questões faz parte da prova.

(2) EXPLIQUE e JUSTIFIQUE todas as suas respostas. Você pode usar os teoremas e algoritmos vistos nas aulas que cobrem os temas da prova. Nesses casos, escreva o que você está usando.

(3) A prova é sem consulta, termina 17:20hs e cada questão vale 20 pontos. Depois da prova o gabarito estará disponível na página web da disciplina.

---

1) *Defina* dois grafos com o mesmo número de vértices e o mesmo número de arestas mas não isomorfos.

$\rightarrow$ $\big(\{1,2,3,4\},\{12,34\}\big)$ *e* $\big(\{1,2,3,4\},\{12,23\}\big)$ *que não são isomorfos pois o primeiro tem todos os vértices de grau 1 e o segundo não.*

2) Decida se é verdadeiro ou falso e justifique: em todo grafo vale a desigualdade triangular

$$\operatorname{dist}(u,w) \leq \operatorname{dist}(u,v) + \operatorname{dist}(v,w)$$

para quaisquer vértices u, v, w do grafo.

$\rightarrow$ *Se* $P_1 = u, x_1, \dots, x_{\operatorname{dist}(u,v)-1}, v$ *é um caminho de comprimento mínimo que liga* u *e* v*, e se* $P_2 = v, y_1, \dots, y_{\operatorname{dist}(v,w)-1}, w$ *é um caminho de comprimento mínimo que liga* u *e* w *então*

$$u, x_1, \dots, x_{\operatorname{dist}(u,v)-1}, v, y_1, \dots, y_{\operatorname{dist}(v,w)-1}, w$$

é um caminho de comprimento $\operatorname{dist}(u,v) + \operatorname{dist}(v,w)$ *que liga* u *a* w*, esse comprimento é pelo menos tão grande quanto a distância* $\operatorname{dist}(u,w)$ *pois, por definição, distância é o comprimento do menor caminho. Se não existe* P_1 *ou não existe* P_2 *a desigualdade fica* $\operatorname{dist}(u,w) \leq \infty$ *o que é verdade qualquer que seja* $\operatorname{dist}(u,w)$*.*

3) Mostre que todo grafo r-regular de cintura 4 tem pelo menos $2r$ vértices.

$\rightarrow$ *Seja* G *um grafo* r*-regular de cintura quatro e tome* v *um vértice qualquer de* G*. A vizinhança* $N_G(v)$ *é um conjunto independente, caso contrário haveria um circuito* C^3 *contrariando a hipótese da cintura. Portanto, cada vértice* $w \in N_G(v)$ *tem que estar ligado a* $r-1$ *vértices de* G *diferentes do vértice* v *e dos vértices em* $N_G(v)$*. Contando o número de vértices,* G *tem pelo menos* $1 + |N_G(v)| + r - 1 = 2r$ *vértices.*

4) Demonstre: se $\delta(G) \geq 3$ então G tem circuito de comprimento par. (*Dica*: comece pelo maior caminho.)

$\rightarrow$ *Seja* $P = x_0, x_1, \dots, x_{k-1}, x_k$ *o maior caminho em* G*. Dessa forma, todos os vizinho de* x_0 *são vértices de* P *e como* $\delta(G) \geq 3$ *o vértice* x_0 *tem pelo menos dois vizinhos diferentes de* x_1*. Sejam* x_i *e* x_j *esses vizinhos,* $j > i > 1$*. Note que temos os circuitos*

$$C^{i+1} = x_0, x_1, \dots, x_i, x_0 \ \text{ e } \ C^{j+1} = x_0, x_1, \dots, x_j, x_0.$$

Se um deles for par, não há mais o que provar, caso contrário, isto é ambos são de comprimento ímpar, então j *é par,* i *é par e o circuito*

$$x_0, x_i, x_{i+1}, \dots, x_{j-1}, x_j, x_0$$

tem comprimento $j - i + 2$ *que é par.*

5) Considere o seguinte algoritmo que recebe um grafo G representado por sua matriz de adjacências e determina se G contém triângulo.

Algoritmo 1 TRIANGULO(G)

PARA $(i = 1; i \leq |V| - 2; i++)$
  PARA $(j = i + 1; j \leq |V| - 1; j++)$
    PARA $(k = j + 1; k \leq |V|; k++)$
      SE $(A[i][j] == A[i][k] == A[j][k] == 1)$
        Devolva 'sim'
Devolva 'não'

(a) Determine a complexidade do algoritmo 1.

→ *$O(|V|^3)$ por causa dos três laços aninhados.*

(b) Seja $A = (a_{ij})$ uma matriz $n \times n$. Aprendemos na escola que o produto $A \cdot A = (b_{ij})$ pode ser calculado por $b_{ij} = \sum_k a_{ik}a_{kj}$. Mostre que são realizadas $O(n^3)$ operações aritméticas para se determinar A^2.

→ *Cada b_{ij} consome $n-1$ somas e n multiplicações, portanto $O(n)$. Como são n^2 posições no total temos $O(n^3)$ operações aritméticas.*

(c) Em 1969, Strassen mostrou um algoritmo que determina o produto de matrizes $n \times n$ em $O(n^{\log 7}) = O(n^{2,81})$ operações aritméticas. Como o algoritmo de Strassen pode ser usado para determinar se um grafo tem triângulo em tempo $O(|V|^{2,81})$? Esse algoritmo é mais eficiente que o algoritmo 1?

→ *Compute A^2, onde A é a matriz de adjacências de G. Para cada $ij \in E(G)$, se $b_{ij} \geq 1$ então o grafo tem triângulo pois b_{ij} é o número de vizinhos que i e j têm em comun. A complexidade é $O(|V|^{2,87})$ (linha 1 do algoritmos 2 abaixo) mais $O(|V|^2)$ (linhas $2 - 6$ do algoritmos 2), portanto, $O(|V|^{2,87})$. Como $n^{2,87} < n^3$ para todo natural $n > 1$ esse algoritmo é mais eficiente.*

Algoritmo 2 TRIANGULO(G)

1: Compute $B = A^2$ usando o algortimo de Strassen;
2: PARA $(i = 1; i \leq |V| - 1; i++)$
3:   PARA $(j = i + 1; j \leq |V|; j++)$
4:     SE $(A[i][j] == 1$ E $B[i][j] \geq 1)$
5:       Devolva 'sim'
6: Devolva 'não'