

PRIMEIRA PROVA DE ALGORITMOS EM GRAFOS
BCC, 1o. SEMESTRE DE 2005

1. [3 pontos] Um (n, m) -grafo é um grafo com n vértices e m arestas. Desenhe, a menos de isomorfismo, *todos* os $(4, 4)$ -grafos (são dois) e *todos* os $(5, 5)$ -grafos.
2. [3 pontos] Suponha que o grafo conexo G tem um vértice de grau k , onde k é um inteiro positivo. Mostre que há em G vértices x_1, \dots, x_k distintos tais que, para todo $1 \leq i \leq k$, o grafo $G - x_i$ é conexo. Descreva um algoritmo para encontrar tais vértices x_i ($1 \leq i \leq k$). A descrição de seu algoritmo pode ser informal, mas deve ser precisa!
3. [4 pontos] Considere a função recursiva

```
void dfsR(Graph G, Edge e)
{ link t; int w = e.w;
  pre[w] = cnt++; st[e.w] = e.v;
  for (t = G->adj[w]; t != NULL; t = t->next)
    if (pre[t->v] == -1)
      dfsR(G, EDGE(w, t->v));
}
```

Seja G o grafo de 13 vértices e 16 arestas dado por

0:	6	5	1		6:	7	2	0	
1:	2	0			7:	10	8	6	
2:	6	1			8:	10	7		
3:	5	4			9:	11	4		
4:	11	9	5	3		10:	8	7	
5:	4	3	0			11:	12	9	4
						12:	11		

Desenhe G . Simule a execução de $\text{dfsR}(G, \{0, 0\})$ (busca em profundidade a partir do vértice 0). Desenhe a árvore de busca em profundidade obtida. Determine os vetores $\text{pre}[]$ e $\text{st}[]$ após a execução desta chamada.

Classifique todas os pares ordenados (a, b) com $ab \in A(G)$ nos quatro tipos (i) tipo árvore, (ii) tipo pai, (iii) tipo ascendente, (iv) tipo descendente.