PRIMEIRA PROVA DE ALGORITMOS EM GRAFOS
BCC, 1o. SEMESTRE DE 2004

1. [3 pontos] Desenhe de forma organizada *todas* as árvores com o conjunto de vértices $\{1, 2, 3, 4\}$. Seja t_n o número total de árvores *não-isomorfas* com n vértices (por exemplo, $t_4 = 2$). Determine t_5 e t_6.
2. [3 pontos] Seja $G = (V, E)$ um grafo. Podemos definir um grafo dirigido $\vec{G} = (V, \vec{E})$ a partir de G, atribuindo para cada aresta $e = \{x, y\}$ de G uma de suas duas possíveis orientações (formalmente, temos $(x, y) \in \vec{E}$ ou $(y, x) \in \vec{E}$, mas não ambos). Podemos chamar um tal $\vec{G}$ de uma *orientação* de G. Suponha que G tem duas árvores geradoras T_1 e T_2 disjuntas nas arestas, isto é, T_1 e T_2 são subárvores de G com $V(T_1) = V(T_2) = V(G)$ e $E(T_1) \cap E(T_2) = \emptyset$. Mostre que G admite uma orientação $\vec{G}$ fortemente conexa, isto é, uma orientação $\vec{G}$ tal que, para quaisquer x e $y \in V(\vec{G})$, há um caminho dirigido de x para y. **Se você usar o Exercício 14 da lista, você deve resolvê-lo na prova! (Há uma solução sem usar este exercício.)**
3. [4 pontos] O que é uma *ponte*? Considere a rotina de identificação de pontes vista em sala (Programa 18.7):

```
void bridgeR(Graph G, Edge e)
{ link t; int v, w = e.w;
  pre[w] = cnt++; low[w] = pre[w];
  for (t = G->adj[w]; t != NULL; t = t->next)
    if (pre[v = t->v] == -1) {
      bridgeR(G, EDGE(w, v));
      if (low[w] > low[v]) low[w] = low[v];
      if (low[v] == pre[v]) { bcnt++; printf("%d-%d\n", w, v); }
    } else if (v != e.v)
      if (low[w] > pre[v]) low[w] = pre[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
```

   Execute o Programa 18.7 acima em G, a partir do vértice 0, destacando sua saída. Determine também `pre[v]` e `low[v]` para todo vértice `v`. Por que o Programa 18.7 funciona? (Argumente de forma sucinta e informal.)