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

---

(1) LEIA COM ATENÇÃO TODA A PROVA.

(2) ENTENDER o enunciado das questões faz parte da prova.

(3) EXPLIQUE todas as suas RESPOSTAS e os ALGORITMOS.

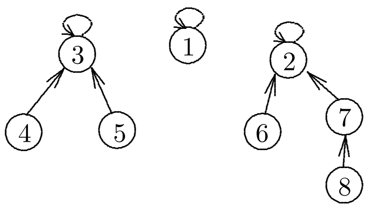
(4) Você pode usar todos os resultados vistos nas aulas. Nesses casos, escreva o que você está usando.

(5) Pode usar lápis e não precisa devolver a folha de questões.

---

1) (**30**) Mostre que se e é uma aresta de menor peso em G então e pertence a alguma árvore geradora mínima. É verdade que e está em toda árvore geradora? Prove ou dê um contra-exemplo.

2) (**30**) Podemos implementar as árvores de subconjuntos disjuntos de um universo $\mathcal{U}$ usando um vetor P indexado pelo conjunto de universo $\mathcal{U} = \{1, 2, \ldots, n\}$ da seguinte forma: o índice do vetor é o conteúdo do nó, e o conteúdo do vetor na posição dada pelo índice é o pai do nó (figura abaixo). Supomos que inicialmente o vetor tem um conteúdo não-válido (0 ou NIL, por exemplo).



P:	1	2	3	3	3	2	2	7	NIL
	1	2	3	4	5	6	7	8	9

Com essa estrutura de dados a escrita dos algoritmos é bem simples:

```
MAKE(v)
  P[v] = v

FIND(v)
  ENQUANTO (v ≠ P[v])
      v = P[v]
  DEVOLVA P[v]

UNION(u, v)
  P[FIND(v)] = FIND(u)
```

(a) Escreva um algoritmo FIND recursivo que faça compressão de caminhos (figura 1).

(b) Suponha que exista mais um vetor T indexado por $\mathcal{U}$. Escreva um algoritmo UNION para união por tamanho em tempo $O(1)$.

3) (**30**) Modifique o Método Húngaro (algoritmo 1) de modo que o algoritmo resultante devolva um emparelhamento máximo em $G = (A \cup B, E)$.

4) (**10**) Seja G um grafo dirigido com peso $\rho\colon E \to \{0, 1, \ldots, W-1\}$ nas arestas, para $W \in \mathbb{N}$ fixo. Escreva uma versão do algoritmo de Dijkstra (algoritmo 2) para computar caminhos mínimos em tempo $O(W|V| + |E|)$.

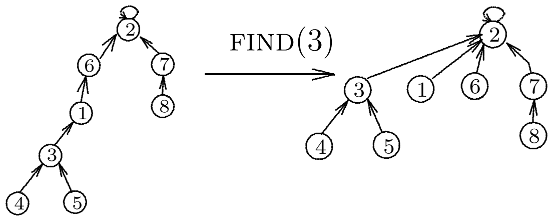


FIGURA 1. Exemplo de FIND com compressão de caminhos.

Algoritmo 1 MÉTODO HÚNGARO(G)

1: Seja M um emparelhamento arbitrário;
2: SE M não cobre A
3:     Escolha $u \in A$ não coberto;
4:     $S = \{u\}$;
5:     $T = \emptyset$;
6: SENÃO
7:     DEVOLVA M;
8: SE $N(S) = T$
9:     DEVOLVA S;
10: SENÃO
11:     Escolha $b \in N(S) - T$;
12: SE $ab \in M$
13:     $S = S \cup \{a\}$;
14:     $T = T \cup \{b\}$;
15:     Volte para 8;
16: SENÃO
17:     Tome P caminho M-aumentante de u até b e faça $M = M \nabla E(P)$;
18:     Volte para 1.

Algoritmo 2 DIJKSTRA(s)

PARA (v=1 ; v$\leq$ |`V`| ; v++)
    d[v] $= \infty$;
    pai[v]=NIL;
d[s]=0;
S $= \emptyset$;
Q=V; `/* Q uma fila de prioridade sobre V=1,...,n com chave d[v], v`$\in$`V`
ENQUANTO (Q $\neq \emptyset$)
    u=EXTRAI-MIN(Q);
    S = S $\cup$ {u};
    PARA (t$\in$adj[u])
        SE (d[t]>d[u]+ρ(u,t))
            d[t]=d[u]+ρ(u,t);
            pai[t]=u;