

MO417 – Análise de Algoritmos

Ata do exercício 22.2-7

Aula do dia 21/05/2009

Redator: Renato Hirata

Livro adotado: Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C.; Algoritmos. Tradução da 2ª edição americana Teoria e Prática. 2002. Página 271

22.2-7 O diâmetro de uma árvore $T = (V, E)$ é dado por

$$\max_{u, v \in V} \delta(u, v);$$

isto é, o diâmetro é a maior de todas as distâncias de caminhos mais curtos na árvore. Forneça um algoritmo eficiente para calcular o diâmetro de uma árvore e analise o tempo de execução de seu algoritmo.

Solução Apresentada

O problema pode ser resolvido da seguinte forma: calcula-se o maior caminho mais curto a partir de um nó arbitrário. O nó mais distante encontrado será uma das extremidades do diâmetro e dessa forma pode ser utilizado como origem de um novo cálculo de um maior caminho mais curto para se obter o diâmetro.

O algoritmo utilizado para se obter o maior caminho mais curto baseia-se no BFS com uma pequena alteração: retornar o ponteiro para o nó mais distante.

Assim, o BFS modificado (BFS2) é aplicado duas vezes.

Algoritmo

```
1 Extremidade1 ← BFS2 (G, s) // s = nó aleatório da árvore
2 Diâmetro ← d [BFS2 (G, Extremidade1)]
```

BFS2 (G, s)

```
1 for cada vértice u ∈ V[G] - {s}
2   do cor[u] ← BRANCO
3     d[u] ← ∞
4     π[u] ← NIL
5 cor[s] ← CINZA
6 d[s] ← 0
7 π[s] ← NIL
8 Q ← vazio
9 ENQUEUE(Q, s)
10 while Q ≠ vazio
11   do u ← DEQUEUE(Q)
12     for cada v ← Adj[u]
13       do if cor[v] = BRANCO
14         then cor[v] ← CINZA
15           d[v] ← d[u] + 1
16           π[v] ← u
17           ENQUEUE(Q, v)
18   cor[u] ← PRETO
19 return u;
```

Complexidade da Solução

A complexidade da solução é 2 vezes a complexidade do BFS, ou seja $2O(V+E) = O(V+E)$; onde V é o número de Vértices e E o número de arestas.

Prova

Ainda não foi apresentada uma prova de correção da Solução. Tal prova foi proposta como um desafio.