

MO417 – Ata de Exercício

Juliana Galvani Gregghi – RA 079782

17 de maio de 2010

Exercício 22.2-7

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

$$\max \delta(u, v) \quad \forall u, v \in V$$

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

Algoritmo *BFS_Diametro* (G, s)

1. para cada vértice $u \in V[G] - \{s\}$ faça
 2. $cor[u] \leftarrow$ BRANCO
 3. $d[u] \leftarrow \infty$
 4. $\pi[u] \leftarrow$ NIL
 5. $cor[s] \leftarrow$ CINZA
 6. $d[s] \leftarrow 0$
 7. $\pi[s] \leftarrow$ NIL
 8. $Q \leftarrow \emptyset$
 9. ENQUEUE(Q, s)
 10. enquanto $Q \neq \emptyset$ faça
 11. $u \leftarrow$ DEQUEUE(Q)
 12. para cada $v \leftarrow Adj[u]$ faça
 13. se $cor[v] =$ BRANCO então
 14. $cor[v] \leftarrow$ CINZA
 15. $d[v] \leftarrow d[u] + 1$
 16. $\pi[v] \leftarrow u$
 17. ENQUEUE(Q, v)
 18. $cor[u] \leftarrow$ PRETO
 19. return ($u, d[u]$)
-

O algoritmo acima é uma versão modificada do algoritmo BFS original, apresentado em (Cormen et

al., 2001, pg.532). A modificação feita encontra-se na linha 19 – o retorno do último vértice alcançado e a distância do mesmo – e se faz necessária para que a solução proposta, apresentada no algoritmo *Diametro_T* descrito abaixo, possa funcionar.

Algoritmo *Diametro_T*(G, s)

1. $(u, d[u]) \leftarrow BFS_Diametro(G, s)$
2. $(v, d[v]) \leftarrow BFS_Diametro(G, u)$
3. return ($d[v]$)

A solução proposta segue a seguinte ideia: ao aplicar-se a busca em largura a um nó qualquer de um grafo, a árvore resultante e, conseqüentemente o último nó alcançado, deve ser um nó de uma das extremidades da árvore e um dos nós mais distante do nó a partir do qual foi iniciada a busca.

Aplicando-se o algoritmo ao nó encontrado, o resultado será, novamente, um nó de uma das extremidades da árvore resultante e, conseqüentemente, a maior de todas as distância de caminhos mais curtos encontrados.

A figura 1 apresenta um grafo G e a solução, passo a passo, da aplicação do algoritmo *BFS_Diametro*(G, s).

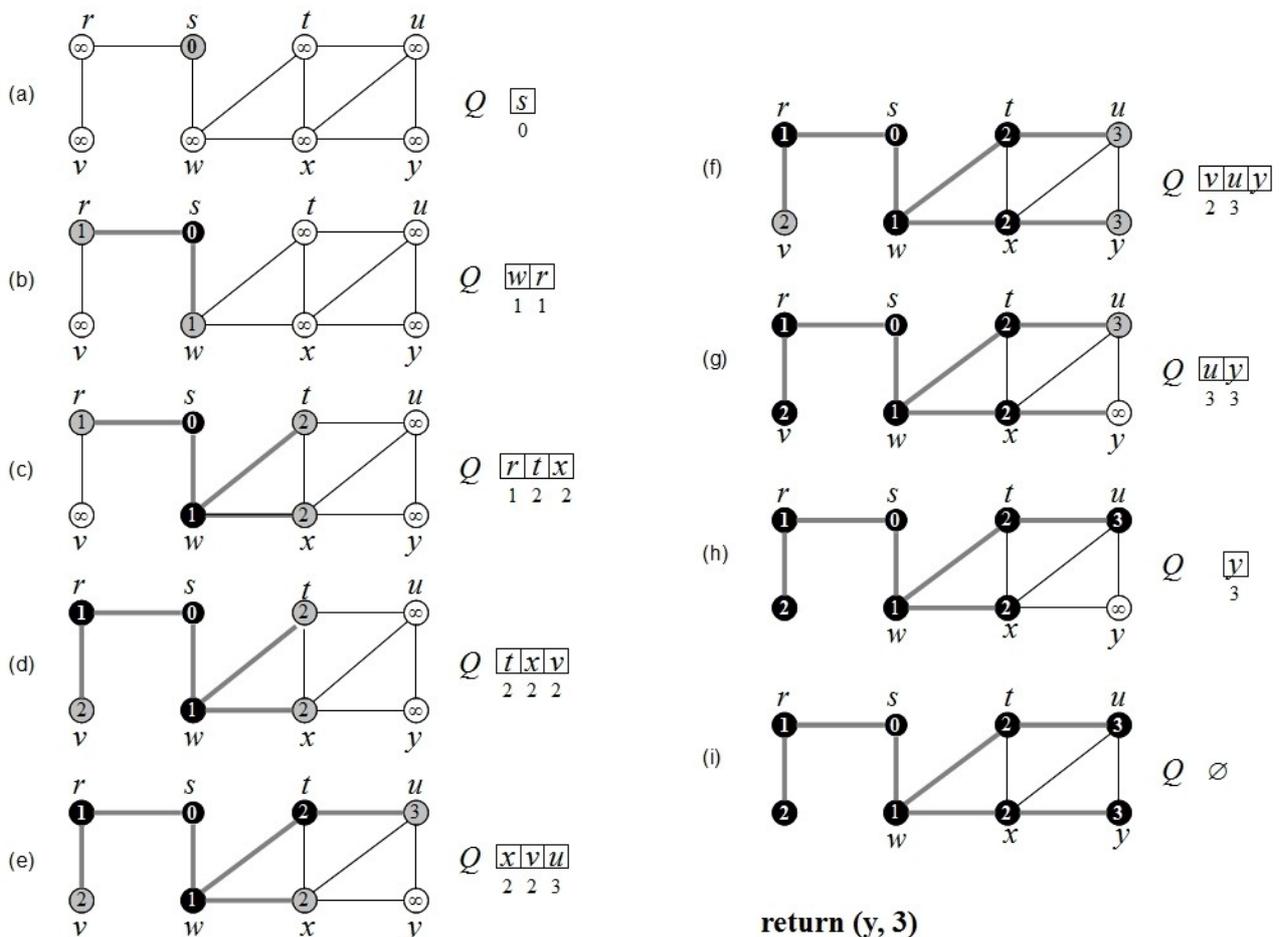


Figura 1 – Execução do algoritmo *BFS_Diametro*(G, s)

Aplicando-se novamente o algoritmo a partir do nó retornado na primeira execução - $BFS_Diametro(G, y)$ - tem-se a seguinte solução (Figura 2).

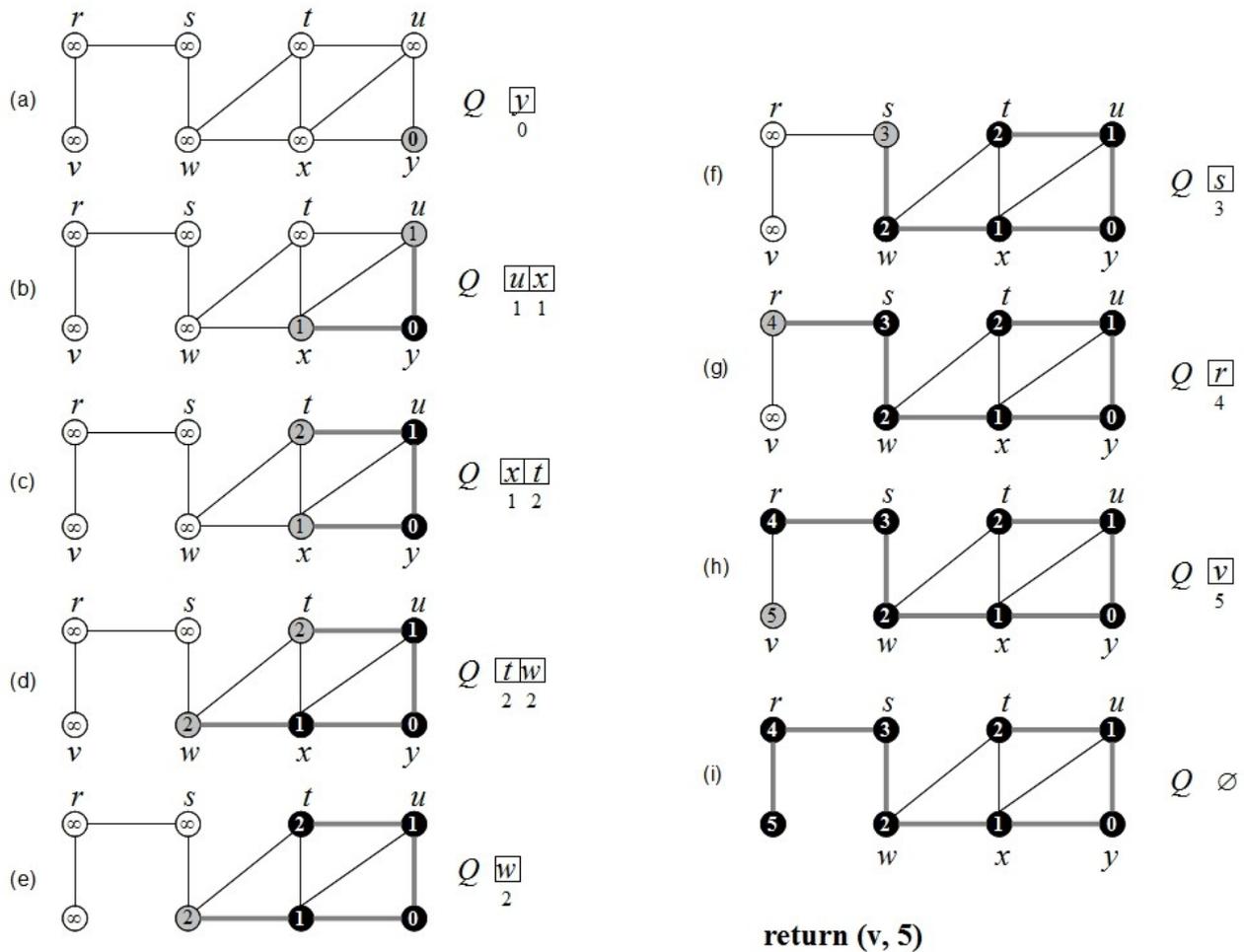


Figura 2 – Aplicação do $BFS_Diametro$ ao nó resultante da primeira execução

O resultado será $d[v]=5$, que é o maior dos caminhos mais curtos encontrados, ou seja, o diâmetro da árvore gerada.

Esta solução tem complexidade igual a 2 vezes a complexidade do algoritmo BFS original, ou seja, tem complexidade $2O(V + E) = O(V + E)$, onde V é o número de vértices em G e E é o número de arestas em G .