

PUC-Rio  
Departamento de Informática  
Prof. Marcus Vinicius S. Poggi de Aragão  
Período: 2008.1  
2 de maio de 2008  
Horário: 2as-feiras e 4as-feiras de 9 às 11 horas

## ANÁLISE DE ALGORITMOS (INF 1721)

### 2ª Lista de Exercícios

1. Exercícios Livro *Algorithms*, Dasgupta, Papadimitriou e Vazirani.

- Cap. 3: 3.7, 3.11, 3.18, 3.19, 3.25, 3.26, 3.27, 3.31;
- Cap. 4: 4.1, 4.3, 4.4, 4.8, 4.10, 4.11, 4.12, 4.13, 4.18.

2. Seja o algoritmo de Dijkstra abaixo.

- Analise sua complexidade explicando as estruturas de dados utilizadas para que a complexidade seja  $O(n^2)$ .
- Analise sua complexidade explicando as estruturas de dados utilizadas para que a complexidade seja  $O(m \log n)$ .
- Explique que alterações devem ser feitas para que este algoritmo encontre uma árvore geradora mínima de  $G$  (algoritmo de Prim).

#### Algoritmo Dijkstra ( $s$ - fonte)

Passo 0: *Inicialização*

Seja  $S$  o conjunto de vértices com o caminho mais curto a partir de  $s$  já determinado, e  $\bar{S}$  seu complemento ( $\bar{S} = V - S$ ).

$S \leftarrow \emptyset$

$d(i) \leftarrow +\infty \forall i \in V$ ;

$d(s) \leftarrow 0$ ;  $pred(s) \leftarrow 0$ ;

Passo 1: *Iteração*

Enquanto  $S \subset V$  faça

1.1 Encontre  $v \in \bar{S}$  t.q.  $d(v) = \min_{w \in \bar{S}} d(w)$

1.2  $S \leftarrow S \cup \{v\}$ ;  $\bar{S} \leftarrow \bar{S} \setminus \{v\}$ ;

1.3 Para todo  $w \in \Gamma^+(v)$

Se  $d(v) + l_{vw} < d(w)$

então  $d(w) \leftarrow d(v) + l_{vw}$ ;  $pred(w) \leftarrow v$ ;

3. Seja  $G = (V, E)$  um grafo orientado e acíclico, com distâncias  $l_e$  para  $e \in E$ , e  $s$  um vértice a partir do qual existe caminho para todos os demais vértices do grafo. Proponha um algoritmo com complexidade  $O(m)$ ,  $m = |E|$ , para encontrar os c.m.c.'s de  $s$  aos outros vértices do grafo. (Dica: use ordenação topológica.)

4. Considere o Algoritmo de Dijkstra para encontrar o Caminho-mais-curto (c.m.c.) entre o vértice  $s$  e os outros vértices de um grafo  $G = (V, E)$ , orientado onde as distâncias dos arcos  $e \in E$  é dada por  $w_e$ .
- Construa um grafo com alguns arcos  $e$  com  $w_e < 0$ , mas sem ciclos de comprimento negativo, onde o algoritmo de Dijkstra funciona **corretamente**.
  - Construa um grafo com alguns arcos  $e$  com  $w_e < 0$ , mas sem ciclos de comprimento negativo, onde o algoritmo de Dijkstra **falha** em encontrar o c.m.c de  $s$  aos outros vértices.
5. Prove que é verdade ou que é falso (nesse caso apresentando um contra-exemplo).
- Se todas as distâncias dos arcos são diferentes, então a árvore de c.m.c. (de  $s$  aos outros vértices do grafo) é **única**.
  - Considere a distância dos c.m.c. de  $s$  aos outros vértices do grafo. Se a distância de cada arco é aumentada de  $k$  unidades, a distâncias dos c.m.c. aumentam de um múltiplo de  $k$ .
  - Se forem retiradas a orientações dos arcos de um grafo orientado  $G$  (i.e. passa ser possível passar nos dois sentidos), as distâncias dos c.m.c.'s permanecem as mesmas.
  - Entre todos os c.m.c.'s existentes entre dois vértices em um grafo, o algoritmo de Dijkstra sempre acha o c.m.c. que possui o menor número de arestas.
6. Seja  $G = (V, E)$  um grafo orientado e acíclico, e  $s$  um vértice a partir do qual existe caminho para todos os demais vértices do grafo. Proponha um algoritmo com complexidade  $O(m)$ ,  $m = |E|$ , para encontrar os c.m.c.'s de  $s$  aos outros vértices do grafo. Para isso, percorra os vértices seguindo uma ordenação topológica.
- Uma ordenação topológica em um grafo orientado acíclico  $G = (V, E)$  é uma ordenação dos vértices do grafo onde se um vértice  $v$  vem antes de um vértice  $w$  então não existe caminho de  $w$  para  $v$ .
  - Para se obter uma ordenação topológica observe que um grafo acíclico tem sempre pelo menos um vértice com grau de entrada igual a ZERO. (Por que ?) Portanto é possível se obter uma ordenação topológica retirando sucessivamente vértices de grau zero, que aparecem na ordenação antes dos que permanecem no grafo.
- O seu algoritmo deve encontrar uma ordenação topológica e o c.m.c de  $s$  aos demais vértices do grafo em  $O(m)$ . Explique com detalhes este algoritmo.
7. Suponha que foram obtidos os c.m.c.'s de  $s$  aos outros vértices de  $G$  e a árvore de c.m.c. é conhecida. Suponha agora que as distâncias de todos os arcos deve ser aumentada de  $k$  unidades. Proponha um algoritmo  $O(m)$  para obter os novos c.m.c.'s.
8. Uma floresta é um grafo sem ciclos. Seja um grafo  $G = (V, E)$ , não-orientado onde os pesos das arestas  $e \in E$  são dados por  $d_e$ . Proponha um algoritmo para encontrar uma floresta com  $k$  arestas,  $k \leq n - 1$ , de peso total mínimo. Sua complexidade tem que ser  $O(n^2)$  ou inferior.
9. Seja o grafo  $G = (V, E)$ , não-orientado onde os pesos das arestas  $e \in E$  são dados por  $d_e$ . Seja também  $T(e)$  a árvore geradora mínima que inclui a aresta  $e$ . Proponha um algoritmo que encontra as árvores geradoras  $T(e)$  para  $e \in E$  (i.e. são  $m = |E|$  árvores geradoras mínimas) (todas em  $O(n^2)$  ( $n = |V|$ )).

10. Considere que a árvore geradora de peso mínimo (AGM) de  $G = (V, E)$ , não-orientado onde os pesos das arestas  $e \in E$  são dados por  $d_e$ , é conhecida. Considere agora que um novo vértice foi acrescentado à  $G$  com arestas para todos os vértices em  $V$ . Proponha um algoritmo para encontrar a nova AGM. Seu algoritmo deve executar em  $O(n \log n)$  ou menos. Tente encontrar um algoritmo  $O(n)$ , existe.
11. Considere um mapa rodoviário e um motorista que tem que ir do vértice  $s$  ao  $t$ . O mapa é representado pelo grafo  $G = (V, E)$ , não-orientado onde os valores associados às arestas  $e \in E$ ,  $h_e$ , correspondem às altitudes das estradas correspondentes aos trechos. O motorista não gosta de altitude e quer fazer o caminho que minimiza a maior altitude que ele vai passar. Utilize um algoritmo de árvore geradora mínima para encontrar esse caminho. Qual a complexidade?