Departamento de Ciência de Computadores FCUP

Desenho e Análise de Algoritmos (CC211) 2013/14

Exame (21.01.2014) *duração: 3h + 30m*

N.º [   ] Nome [   ]

1. Pretendemos relacionar as classes $O(3\log(n^2)+5n)$, $\Omega(\log n)$, $\Theta(n^2\log n)$ e $O(n)$.

a) Complete a tabela, escolhendo o símbolo **mais adequado** de $\{\subseteq, \not\subseteq, \subset, \not\subset, =\}$ para cada entrada.

	$O(3\log(n^2)+5n)$	$\Omega(\log n)$	$\Theta(n^2\log n)$	$O(n)$
$O(3\log(n^2)+5n)$	$=$			
$\Omega(\log n)$		$=$		
$\Theta(n^2\log n)$			$=$	
$O(n)$				$=$

b) Recorrendo às definições das notações Θ, O e Ω, justifique formalmente a resposta que deu para o par $(O(3\log(n^2)+5n), O(n))$.

2. A função Inverte(v,a,b) inverte o segmento $v[a], v[a+1], \ldots, v[b]$ de um vetor v de n elementos, **quando** $0 \leq a \leq b < n$. Se o estado inicial for $(v_a, v_{a+1}, \ldots, v_{b-1}, v_b)$ então, após a chamada da função, o estado desse segmento é $(v_b, v_{b-1}, \ldots, v_{a+1}, v_a)$.

```
Inverte(v, a, b)
    k ← 0;
    Enquanto (a + k < b − k) fazer
        aux ← v[a + k];
        v[a + k] ← v[b − k];
        v[b − k] ← aux;
        k ← k + 1;
```

a) Prove formalmente a correção de Inverte(v,a,b). Deve usar indução sobre o número de vezes que a condição $a+k<b-k$ é testada.

b) Admita que cada uma das instruções de atribuição tem **duração unitária**, bem como o teste $a+k<b-k$ e as transferências de controlo no ciclo (após esse teste e no fim do bloco de instruções, em cada iteração). Deduza a expressão que define o tempo $T(a,b)$ que Inverte(v,a,b) demora, e caraterize a sua complexidade assintótica.

3. Recorde que uma das versões dadas do **algoritmo** de Floyd-Warshall inclui o fragmento seguinte.

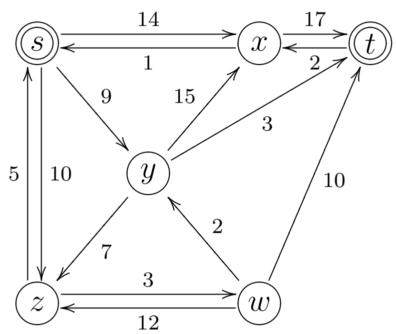
```
Para i ← 1 até n fazer
   Para j ← 1 até n fazer
      Se D[i, j] > D[i, k] + D[k, j] então
         D[i, j] ← D[i, k] + D[k, j];
         P[i, j] ← k;
```

a) Apresente o problema que esse algoritmo resolve e a complexidade temporal e espacial do algoritmo.

b) Como é que o fragmento indicado contribui para a resolução do problema? Qual é a ideia? Que propriedade é crucial para a correção deste fragmento?

(Continua, v.p.f.)

4. Seja $G = (V, E, c, \{s, t\})$ a rede representada, com origem s e destino t, sendo a capacidade $c(e)$ o valor indicado no ramo e, para todo $e \in E$. Considere o fluxo f em G tal que $f(u, v) > 0$ apenas para os pares $(u, v) \in V \times V$ apresentados à direita e tem os valores indicados para esses pares.



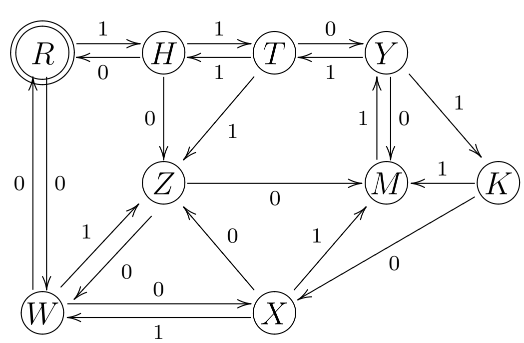
$$
\begin{array}{rclcrcl}
f(s,x) & = & 10 & \quad & f(y,x) & = & 7 \\
f(s,y) & = & 9 & & f(y,t) & = & 1 \\
f(s,z) & = & 1 & & f(y,z) & = & 2 \\
f(w,t) & = & 2 & & f(z,w) & = & 3 \\
f(w,y) & = & 1 & & f(x,t) & = & 17
\end{array}
$$

a) Apresente as definições relevantes para o cálculo da rede residual G_f associada a f e represente-a.

b) Justifique que $|f|$ não é o valor do fluxo máximo e, **partindo do fluxo** f, determine um fluxo máximo em G por aplicação do algoritmo de Edmonds-Karp. Explique, indicando os passos principais do algoritmo, de forma clara e sucinta, e apresentando o valor do fluxo após cada iteração, nos pares $(u, v) \in V \times V$ alterados.

5. Seja $G = (V, E, p)$ um grafo dirigido que serve de modelo a uma rede urbana e em que $p(e) \in \{1, 0\}$ indica se o ramo $e \in E$ está ou não congestionado (sendo 1 se estiver). Para um nó fixo $s \in V$, pretendemos encontrar, para todos os $v \in V \setminus \{s\}$, um caminho de s para v, sem congestionamentos ou com no máximo um ramo congestionado. Cada um desses caminhos deve ser o mais curto possível (isto é, ter o menor número de ramos). Em nenhum caso se admitirão caminhos com dois ou mais ramos congestionados. Se o caminho mais curto tiver algum congestionamento, então deverá também ser obtido o melhor caminho sem congestionamentos (caso exista). Em caso de empate, pode ser indicado um qualquer. Para a caraterização de um tal caminho até v, deve ser indicado um **terno**: se tem ou não congestionamento, qual é o seu comprimento e qual é o nó que antecede v nesse caminho.

a) Por aplicação de uma estratégia baseada em pesquisa em largura, resolva a instância seguinte, com origem $s = R$. Na resolução, deve indicar resultados intermédios que permitam compreender o algoritmo que está a aplicar, o qual deverá ter complexidade temporal $O(|E| + |V|)$.



b) Traduza (em pseudocódigo) o algoritmo que concebeu para resolver o problema em tempo $O(|E| + |V|)$, sendo s dado. Como anteriormente, para cada $v \in V$, poderá ter de caraterizar dois caminhos se não forem equivalentes (em termos de distância ou do número de congestionamentos são ambos ótimos). Por isso, é útil saber se o resultado obtido para v corresponde a 0, 1 ou 2 caminhos.

c) Justifique sucintamente a correção do algoritmo e a sua complexidade temporal.

(Continua, v.p.f.)

N.º | | Nome | |

Das perguntas 6, 7 e 8, deve resolver apenas duas

(se resolver mais, serão classificadas apenas as perguntas 6 e 7)

6. Seja G um grafo dirigido finito, com n vértices, numerados de 1 a n, com valores $d(v,w) \in \mathbb{Z}^+$, para todo o ramo (v,w). Considere o fragmento seguinte de uma função que traduz uma das versões do algoritmo de Dijkstra dadas. Assuma que a fila de prioridade Q é suportada por uma *heap binária de mínimo* como a que foi **descrita e usada nas aulas**.

```
Enquanto (PQ_NOT_EMPTY(Q)) fazer
   v ← EXTRACTMIN(Q);
   Se (v = t ∨ dist[v] = ∞) então retorna;
   Para cada w ∈ Adjs[v] fazer
      Se dist[v] + d(v,w) < dist[w] então
         dist[w] ← dist[v] + d(v,w);
         pai[w] ← v;
         DECREASEKEY(Q, w, dist[w]);
```

1 : $(2,3),(8,10)$
2 : $(1,6),(3,3),(4,10),(7,8),(8,17)$
3 : $(4,1)$
4 : $(1,2),(3,7),(5,2),(6,1),(8,1)$
5 : $(4,3)$
6 : $(3,2),(5,1),(7,8)$
7 : $(2,9),(3,4),(6,12)$
8 : $(1,5),(4,4)$

a) Considere a instância de G esquematizada acima, à direita, com $n = 8$ e $t = 5$. Em cada linha, tem a lista de adjacentes de um nó v, pela mesma ordem em que se encontra na estrutura de dados que representa G. Cada par $(w, d(v,w))$ nessa lista identifica o extremo final e o valor de um ramo. Para essa instância, o estado da fila de prioridade Q após uma certa iteração do ciclo "Enquanto" é o seguinte (onde, $v : y$ indica que v tem chave y e as posições vazias têm valores irrelevantes).

a		4 : 4	1 : 6	7 : 8	8 : 17	5 : ∞	6 : ∞		

pos_a		2	0	0	1	5	6	3	4

sizeMax = 8 size = 6

Qual é o estado da fila de prioridade imediatamente após a execução de $v \leftarrow$ EXTRACTMIN(Q) na iteração seguinte? Qual é o estado da fila de prioridade imediatamente após cada operação DECREASEKEY nessa iteração? Qual foi o vértice s (origem) de que se partiu no algoritmo de Dijkstra? Justifique as respostas.

b) Qual é a complexidade temporal assintótica do fragmento indicado? Justifique sucintamente.

7. Considere o problema da determinação de uma árvore geradora mínima de um grafo **conexo** $G = (V, E, p)$, em que $p(e) \in \mathbb{Z}^+$ define o peso do ramo $e \in E$.

a) Para cada um dos três casos, **dê um exemplo ou justifique a não existência** de uma instância, com $|V| = 8$ e $|E| = 10$, em que $p(e) \neq p(e')$ se $e \neq e'$, para todo $\{e, e'\} \subset E$, e tal que:

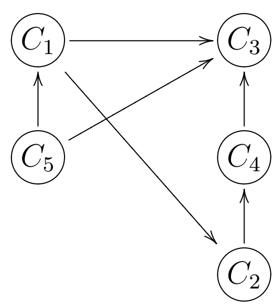
1. nenhuma árvore geradora mínima tem peso igual à soma dos pesos dos sete ramos mais leves.
2. alguma árvore geradora mínima tem peso igual à soma dos pesos dos cinco ramos mais leves.
3. os algoritmos de Prim e de Kruskal produzem árvores distintas (embora com o mesmo peso).

(Continua, v.p.f.)

b) Recorde que qualquer que seja a árvore geradora mínima T de G e qualquer que seja a partição $\{A, B\}$ de V, a árvore T tem algum ramo $\langle a, b\rangle \in E$, com $a \in A$ e $b \in B$, e tal que $p(a, b)$ é igual ao mínimo de $\{p(x, y) \mid x \in A, y \in B, \langle x, y\rangle \in E\}$. Como é que tal propriedade permite concluir que os algoritmos de Prim e de Kruskal efetuam escolhas seguras (ou seja, que conduzem a uma árvore geradora de G e que essa árvore tem peso mínimo)? Na explicação, analise os dois casos possíveis:

- Caso 1: os pesos nos ramos são todos distintos.
- Caso 2: os pesos nos ramos não são todos distintos.

8. Seja $G = (V, E)$ um grafo dirigido, com $V = \{v_i \mid 1 \leq i \leq 17\}$, e tal que que a relação definida por G é irreflexiva e assimétrica, isto é, G não tem lacetes e se $(x, y) \in E$ então $(y, x) \notin E$, qualquer que seja (x, y). Sabe-se ainda que G tem exatamente cinco componentes fortemente conexas, designadas por C_1, C_2, C_3, C_4 e C_5, cada uma com pelo menos três vértices, o grafo das componentes fortemente conexas de G é o que se encontra a seguir e $v_1 \in C_1$, $v_2 \in C_2$, $v_3 \in C_3$, $v_4 \in C_4$ e $v_5 \in C_5$.



a) Para cada um dos três casos seguintes, **dê um exemplo ou justifique a não existência** de um grafo G nas condições indicadas acima e tal que:

1. $\{(v_5, v_1), (v_7, v_5), (v_7, v_1)\} \subset E$;
2. $\{(v_1, v_3), (v_1, v_5)\} \subset E$;
3. no algoritmo de Kosaraju-Sharir, as componentes fortemente conexas de G são reportadas pela ordem C_1, C_2, C_3, C_4, C_5.

b) Apresente os passos principais do algoritmo de Kosaraju-Sharir, de modo que a complexidade temporal do algoritmo seja $\Theta(|V| + |E|)$. Para a complexidade temporal, que importância têm as estruturas de dados utilizadas?

(Fim)

Cotação:

1a) 1	1b) 1.5	
2a) 1.5	2b) 1.5	
3a) 0.5	3b) 1.5	
4a) 1	4b) 1.5	
5a) 1	5b) 2	5c) 1
6a) 2	6b) 1	
7a) 1.5	7b) 1.5	
8a) 1.5	8b) 1.5	