Departamento de Ciência de Computadores FCUP

Desenho e Análise de Algoritmos (CC211) 2013/14

Exame recurso (05.02.2014) *duração: 3h+30m*

Cotação: 2, 3, 3, 3, 3, 3, 3 (3)

N.º ______ Nome ______

1. Usando a definição das classes $\Theta(n^2)$, $O(n\log_2 n)$ e $\Omega(5n + n\log_2 n)$, prove que:

a) $4n^2 + 5n - 8 \in \Theta(n^2)$

b) $O(n\log_2 n) \cap \Omega(5n + n\log_2 n) \neq \emptyset$

2. Considere as duas funções seguintes para resolução do problema da determinação do máximo de um vetor v de n inteiros e do seu número de ocorrências, sendo v indexado a partir de 0.

```
VersaoA(v, n)
  max ← v[0];
  nmax ← 1;
  i ← 1;
  Enquanto (i < n) fazer
      Se max < v[i] então
          max ← v[i];
          nmax ← 1;
      senão se max = v[i] então
          nmax ← nmax + 1;
      i ← i + 1;
  retorna (max, nmax);
```

```
VersaoB(v, n)
  max ← v[0];
  i ← 1;
  Enquanto (i < n) fazer
      Se max < v[i] então
          max ← v[i];
      i ← i + 1;
  nmax ← 0;
  i ← 0;
  Enquanto (i < n) fazer
      se max = v[i] então
          nmax ← nmax + 1;
      i ← i + 1;
  retorna (max, nmax);
```

a) Enuncie um invariante de ciclo que permita justificar formalmente a correção da função `VersaoA`. Demonstre esse invariante por indução matemática. Explique ainda como é que a partir desse invariante se deduz a correção da função.

b) Enuncie os invariantes de ciclo relevantes para a prova de correção da função `VersaoB`.

3. Na continuação da questão 2., suponha que nestas funções todos os testes, transferências de controlo e atribuições (mesmo envolvendo operações) têm duração de **uma unidade** de tempo. Designe por $T_A(v, n)$ e $T_B(v, n)$ os tempos de execução das funções quando aplicadas a uma instância (v, n), contabilizados a partir da instrução $max \leftarrow v[0]$.

a) Justifique que o *contributo global conjunto* das instruções $nmax \leftarrow nmax + 1$ e $nmax \leftarrow 1$ para $T_A(v, n)$ nunca excede n unidades de tempo, qualquer que seja (v, n). Usando esse resultado, justifique que $T_A(v, n) < T_B(v, n)$, qualquer que seja a instância (v, n).

b) Determine minorantes e majorantes para os tempos de execução, válidos para qualquer par (v, n) e adequados para concluir que $T_A(v, n) \in O(n)$ e $T_B(v, n) \in \Omega(n)$. Explique sucintamente.

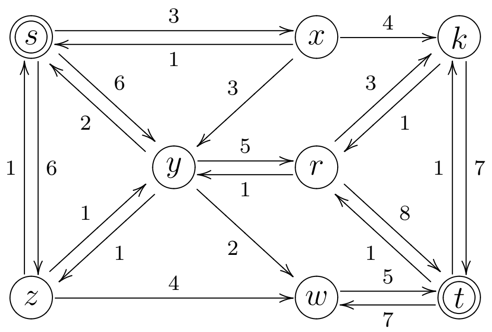
c) Caraterize a complexidade temporal de cada função e averigue se alguma resolve o problema descrito na questão 2. em tempo ótimo do ponto de vista da complexidade assintótica. Justifique a resposta.

(Continua, v.p.f.)

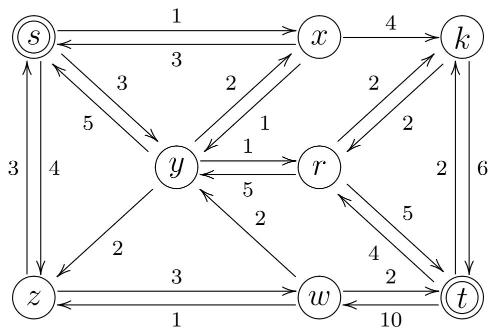
Das perguntas 4–8, deve responder a quatro
(se resolver mais, a pergunta 8 não será classificada)

4. Seja $\mathcal{R} = (V, E, c, \{s, t\})$ uma rede e seja $\mathcal{R}_f = (V, E, c_f, \{s, t\})$ a rede residual determinada por um fluxo f em $\mathcal{R}$. Considere a instância esquematizada.

$\mathcal{R}$:



$\mathcal{R}_f$:



a) Justifique que $\gamma = (s, y, z, w, t)$ é um *caminho para aumento* de f e que podia ser usado na iteração seguinte do método de Ford-Fulkerson mas não na iteração seguinte do algoritmo de Edmonds-Karp.

b) Para os pares de $V \times V$ que seriam afetados pela saturação de $\gamma = (s, y, z, w, t)$, indique o fluxo f e o fluxo após essa alteração. Explique.

c) Seja $\hat{f}$ o fluxo obtido no algoritmo de Edmonds-Karp para melhoramento do **fluxo** f, realizando apenas **quatro** iterações. Indique o valor de $|\hat{f}|$ e averigue se $\hat{f}$ é um fluxo máximo em $\mathcal{R}$. Na justificação, não apresente as redes intermédias mas indique os caminhos usados para aumento. Se houver alternativas, admita que cada nó visitaria os seus adjacentes por ordem alfabética.

5. Considere o fragmento seguinte de uma versão adaptada do algoritmo de Prim para determinação de uma árvore geradora **máxima** de um grafo conexo $G = (V, E, d)$.

> Para cada $w \in Adjs[v]$ fazer
>   Se $d(v, w) > dist[w]$ então
> $dist[w] \leftarrow d(v, w)$;
> $pai[w] \leftarrow v$;
>   IncreaseKey$(Q, w, dist[w])$;

a) Que interpretação tem $pai[x]$ e $dist[x]$, para cada $x \in V$? Como são inicializados, se a raíz escolhida for s? De que modo a alteração realizada neste fragmento preserva essa interpretação?

b) Numa implementação em que Q é suportada por uma *heap binária de máximo* como a descrita nas aulas, o que faz a chamada IncreaseKey$(Q, w, dist[w])$? Apresente a sua complexidade em função de Q.`size`.

Ilustre, explicando o que ocorre na instância seguinte quando IncreaseKey é chamada com $w = 11$, $dist[w] = 40$ e o estado de Q é o indicado abaixo (onde $\alpha : \beta$ identifica o nó e a chave).

Q.`a`		5 : 23	4 : 7	10 : 8	8 : 4	1 : 6	6 : 0	7 : 8	11 : 0	9 : 4		

Q.`sizeMax` = 11 Q.`size` = 9 Q.`pos_a`

	?	?	$\cdots$	?

Se Q não sofresse outras alterações, até ao fim do ciclo "Para" indicado, qual seria o próximo vértice a extrair da fila? Como ficaria Q após a extração? Explique. Apresente também o estado de Q.`pos_a`.

(Continua, v.p.f.)

N.º ____ Nome ____

6. Seja $G = (V, E, t)$ um grafo dirigido finito, com $V = \{1, 2, \ldots, n\}$, que serve de modelo a uma rede de transportes, e em que $t(e) \in \mathbb{R}^+$ define a temperatura no ramo e, para todo $e \in E$. Seja $T(\gamma) = \text{maximo}\{t(e) \mid e \in \gamma\}$, para cada percurso γ. Para todos os pares $(i, j) \in V \times V$, com $i \neq j$, pretendemos encontrar **um percurso** γ_{ij} **tal que** $T(\gamma_{ij})$ **seja mínimo** quando considerados todos os percursos possíveis de i para j.

a) Seguindo uma abordagem de programação dinâmica, apresente uma recorrência que permita calcular $T(\gamma_{ij})$ para todos os pares (i, j), nas condições do enunciado, bem como o segundo vértice no percurso γ_{ij} escolhido. Explique a ideia subjacente e justifique a sua correção.

b) Tendo por base a recorrência que definiu e usando pseudocódigo, escreva um algoritmo polinomial para resolver o problema, com complexidade espacial $O(n^2)$. Indique a sua complexidade temporal.

7. Pretende-se uma função $\textsc{Caminho}(x, y, s, pai, dist, cam)$ para obter o *caminho do nó x para o nó y* existente na árvore geradora escolhida pelo algoritmo de Prim para um grafo G conexo, sendo x e y dados e $x \neq y$. Também *pai* e *dist* já têm os valores finais encontrados por aplicação do algoritmo de Prim (com raíz s). A função deve retornar ainda *a soma dos pesos* dos ramos usados e ter complexidade $O(|V|)$, sendo *cam* o vetor em que ficará o caminho.

a) Escreva a função em pseudocódigo. Na procura do caminho, a árvore deve ser entendida como um subgrafo **não dirigido**. Na função não pode usar explicitamente G (apenas x, y, *dist*, *pai* e s, se necessário). Note que se pretende um caminho (que, por definição, terá de ser um percurso acíclico).

b) Justifique a correção da função que apresentou, indicando a propriedade que explora.

8. Elabore uma exposição sucinta sobre o algoritmo de Kosaraju-Sharir, apresentando o problema que resolve (e dizendo em que consiste), descrevendo os passos principais do algoritmo, e explicando as propriedades que suportam a sua correção e complexidade. Nessa explicação, deverá também referir as propriedades relevantes da pesquisa em profundidade, incluindo as exploradas na ordenação topológica de DAGs.

(Fim)