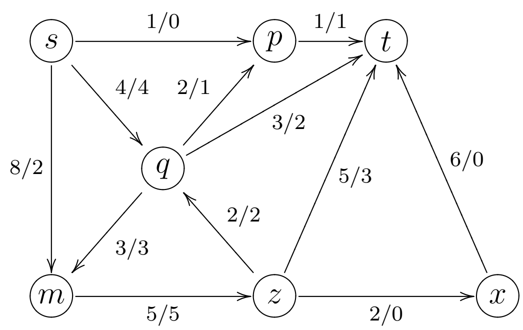
Departamento de Ciência de Computadores — FCUP

Desenho e Análise de Algoritmos (CC2001) 2017/18

Exame (09.01.2018) — *duração: 3h*

N.º ____ Nome ____

1. Considere a rede de fluxo seguinte, onde c/f são pares capacidade/fluxo, e s e t são a origem e destino.



a) [0.7] Indique os valores de:

$f(m,q)$		$f(q,m)$		$f(t,z)$	
$\lvert f \rvert$		$c(q,z)$		$c(z,q)$	
$c_f(q,m)$		$c_f(m,q)$		$c_f(x,z)$	
$c_f(z,x)$		$c_f(s,q)$		$c_f(q,s)$	

b) [0.4] Indique um corte $\{S,T\}$ com capacidade mínima. Qual é a essa capacidade?

c) [1.4] Partindo de f, aplique o algoritmo de Edmonds-Karp para obter um fluxo máximo (desenhe a rede residual **em cada iteração**, represente o fluxo final na rede, e explique sucintamente).

d) [0.3] Qual é a diferença principal entre o método de Ford-Fulkerson e o algoritmo de Edmonds-Karp?

(Continua, v.p.f.)

2. Considere o problema de formar uma certa quantia de q euros e c cêntimos com moedas de valores 1, 2, 5, 10, 20 e 50 cêntimos, 1 e 2 euros, e ainda notas de 5, 10 e 20 euros. Pretendemos usar o número mínimo de notas e moedas. Admita que pode dispor de um **número ilimitado** de notas/moedas de cada tipo.

a) [1.2] Usando pseudocódigo, defina uma função QUANTIA(v, n, q, c, s), com **complexidade** $O(n)$, que determine no *array* s a solução obtida pelo algoritmo *greedy* dado e retorne o número de moedas/notas usadas. O *array* v define o valor das moedas/notas e n o número de tipos.

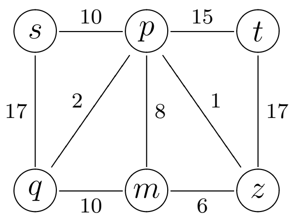
b) [0.2] Se $q = 327$ e $c = 77$, o estado final de s e o valor de retorno são:

c) [0.5] A complexidade de QUANTIA(v, n, q, c, s) é $\Theta(n)$? Justifique.

d) [0.6] Prove que se o número de moedas/notas **for limitado**, a estratégia *greedy* (adaptada) não é correta. Indique todos os erros possíveis (e instâncias correspondentes).

N.º | Nome

3. [1.5] Aplique o algoritmo de Prim para obter uma árvore geradora $\mathcal{T}$ de peso **máximo** do grafo indicado, com **raiz** q. **Em cada iteração**, apresente os nós em $\mathcal{T}$ e o vetor $pai[\cdot]$ e $dist[\cdot]$, como se definiu nas aulas.



4. [0.6] No algoritmo de Dijkstra, suportado por uma *heap binária de mínimo*, para determinação de caminhos mínimos com origem num nó s de um grafo dirigido $G = (V, E, d)$, com $d(e) \in \mathbb{Z}^+$, qual é o estado de $dist[v]$ e $pai[v]$, no fim de cada iteração, para todo $v \in V$? Qual é o invariante de ciclo que garante a correção do algoritmo?

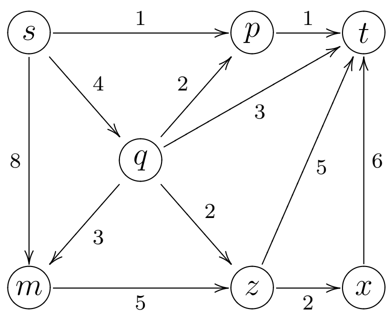
5. [2.0] Usando diretamente a **definição** das ordens de grandeza indicadas:

a) prove que $200n + 5n\log_2 n \in O(3n^2 \log_2 n)$

b) diga, justificando, se $3n + 100 \in \Omega(n^2/3)$.

6. Considere um grafo dirigido **acíclico** $G = (V, E, d)$, com valores com $d(e) \in \mathbb{Z}^+$ associados aos ramos. Cada ramo e representa uma tarefa de um projeto e o valor $d(e)$ representa a sua duração. Todas as tarefas terão de ser realizadas e algumas podem decorrer **simultaneamente**. As tarefas com origem num nó só podem começar depois de todas as tarefas com fim nesse nó estarem concluídas. Admitindo que pode iniciar o projeto no instante 0 (zero), pretendemos determinar o instante em que estaria concluído e o instante em que daria início às tarefas com origem em cada nó se as realizasse o mais cedo possível.

a) [0.5] Para a instância representada, indique o instante em que daria início às tarefas com origem em cada nó. Quando é que o projeto estaria concluído?



b) [3.0] Usando pseudocódigo, apresente um **algoritmo** com complexidade temporal $\Theta(|V| + |E|)$ para resolver o problema. Justifique sucintamente a **correção** e **complexidade** do algoritmo.

7. [0.6] Indique a estratégia *greedy* que o resolve o problema da mochila fracionário (*linear knapsack*).

N.º | Nome

8. [1.5] Apresente a recorrência que serve de base ao algoritmo de Floyd-Warshall, e que define a distância mínima de s para t, para **todos os pares** $(s,t) \in V \times V$, num grafo dirigido pesado $G = (V, E, d)$, com $d(e) \in \mathbb{Z}^+$, para todo $e \in E$, com $V = \{1, 2, \ldots, n\}$. **Explique sucintamente a ideia subjacente**.

9. [2.0] Considere uma *heap binária de mínimo* com 10 elementos, dada por $[-8, -5, 2, -3, 4, 9, 6, 1, 10, 8]$.

a) Indique os valores de: PARENT(7) LEFT(7) RIGHT(7)

b) Represente-a por uma árvore.

c) Desenhe-a após EXTRACTMIN.

d) Desenhe-a após mudar para -13 a chave 1, por DECREASEKEY.

10. [1.0] Qual é a complexidade de uma operação de procura de um elemento com uma chave k dada, numa *heap binária de mínimo*, com n elementos (todos com chaves distintas)?

E, numa árvore de pesquisa *red-black* , num *array* ,

num *array* ordenado ?

(Continua, v.p.f.)

11. Recorde o problema "Caixotes de morangos", em que é necessário determinar como distribuir c caixas de morangos por l lojas de forma a maximizar o valor total obtido. Seja R_{kn} o valor que a loja k oferece por n caixas e seja T_{kn} o valor máximo que se pode obter se se distribuir n caixas pelas lojas $1, 2, \ldots, k$. Seja S_{kn} uma solução com valor T_{kn}, dada por uma lista de pares (q, i), em que q é o número de caixas que envia à loja i, com $q \neq 0$ (omite o par se $q = 0$). Seja N_{kn} o número total de soluções com valor T_{kn}. Assuma que os valores R_{kn} são inteiros positivos.

a) [1.2] Adaptando a função dada nas aulas, escreva (em pseudocódigo) a função $\textsc{Caixotes}(R, c, l, T, S, N)$ para obter os valores T_{ln}, S_{ln} e N_{ln}, **usando programação dinâmica**, para $0 \leq n \leq c$, sendo T e N *arrays* de inteiros, com $c+1$ posições e S um *array* de $c+1$ listas de pares de inteiros. Admita que R é uma matriz de inteiros com l linhas e $c+1$ colunas.

b) [0.5] Justifique sucintamente a correção.

c) [0.3] Indique a complexidade temporal.

(Fim)