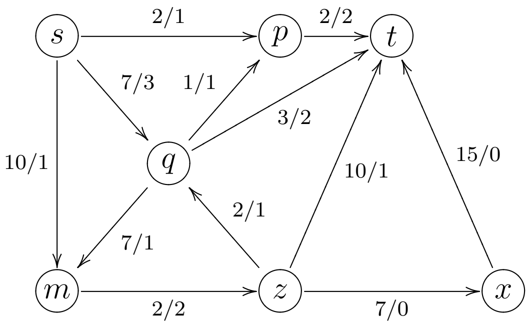
Desenho e Análise de Algoritmos (CC2001) 2017/18

2ºTeste (13.12.2017) *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:

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

b) [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).

c) [0.4] A partir das estruturas de dados calculadas, como se pode identificar um corte $\{S,T\}$ de capacidade mínima? Indique-o e a sua capacidade.

d) [0.4] Sabendo que, para uma rede com n nós e m ramos, o algoritmo não efetua mais do que $mn/2$ iterações, justifique a complexidade $O(m^2n)$.

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) [1.0] Usando pseudocódigo, apresente uma função QUANTIA(v, n, q, c, s) que determine no *array* s a solução obtida pelo algoritmo *greedy*. O *array* v define o valor das moedas/notas e n o número de tipos.

b) [0.4] A complexidade (para v e n quaisquer) é:

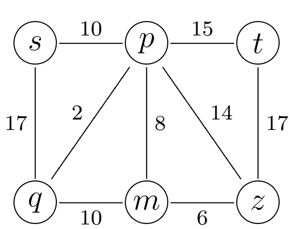
c) [0.1] Na chamada, o estado de v e n é:

d) [0.3] Se $q = 217$ e $c = 79$, o estado final de s é:

e) [0.4] Justifique que QUANTIA(v, n, q, c, s) determina a solução ótima, estendendo a prova de que a estratégia *greedy* produz a solução ótima se se usar apenas moedas (dada nas aulas).

f) [0.5] Prove que se o número de moedas/notas **for limitado**, a estratégia *greedy* (adaptada) não é correta.

3. [1.4] Aplique o algoritmo de Kruskal para obter uma árvore geradora $\mathcal{T}$ de peso **máximo** do grafo indicado. Em cada iteração, apresente os ramos em $\mathcal{T}$ e o conjunto de vértices que constitui cada componente.



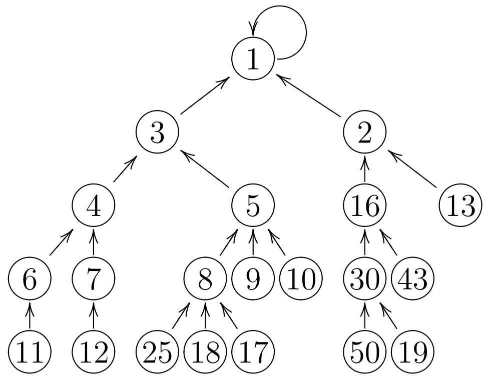
(Continua, v.p.f.)

N.º Nome

4. [1.8] Usando **a definição** das classes prove que $200n + 1000 \in O(n^2 \log_2 n)$ e $200n + 1000 \notin \Omega(n^2)$ e diga, justificando, se se pode concluir que $200n + 1000 \notin \Theta(n^2)$.

5. [3.0] Seja $G = (V, E, c)$ um grafo não dirigido, com $c : E \to \mathbb{Z}^+$ **constante**. Queremos um percurso γ de um nó s para um nó t, com $\sum_{e\in\gamma} c(e) \leq cmax$, para $cmax \in \mathbb{Z}^+$ dado. Apresente em pseudocódigo uma função $\textsc{Caminho}(G, s, t, cmax)$, com complexidade $O(|V| + |E|)$, para **obter** um tal percurso, se existir, e o **imprimir**. Pode usar funções auxiliares. Justifique sucintamente a **correção** e **complexidade**.

6. [0.5] Admita que a árvore representa um dos conjuntos de uma partição de $V = \{1, 2, \ldots, 50\}$. Desenhe a árvore após a operação $\textsc{FindSet}(18)$, supondo que usa a heurística *path compression*.



7. Seja G um grafo dirigido $G = (V, E)$ com $V = \{v_1, v_2, v_3, \ldots, v_{10}\}$, $|V| = 10$, $|E| = 15$ e tal que G tem **exatamente quatro componentes fortemente conexas, duas com dois nós e duas com três nós**. O nó v_1 é acessível de v_5 e de v_9, mas nem v_5 nem v_9 são acessíveis de v_1, o nó v_5 não é acessível de v_9 nem v_9 de v_5, o nó v_3 é acessível de v_1 mas v_1 não é acessível de v_3, e o nó v_7 é acessível de v_3 e de v_9.

a) [0.5] Dê exemplo de um grafo nas condições indicadas e identifique as componentes.

b) [1.2] Assumindo que se houver alternativa num passo da pesquisa, explora primeiramente o nó de índice menor, indique a ordem pela qual as componentes são obtidas no algoritmo de Kosaraju-Sharir. Justifique.

8. [2.0] Considere uma *heap binária de mínimo* com 10 elementos, dada por $[-8, -5, 1, -4, 3, 8, 6, 2, 10, 7]$.

a) Indique os valores de: PARENT(5) LEFT(5) RIGHT(5)

b) Represente-a por uma árvore.

c) Desenhe-a após EXTRACTMIN.

d) Desenhe-a após a operação DECREASEKEY reduzir 7 para -7.

e) Na definição dada nas aulas para uma fila de prioridade suportada por uma *heap binária de mínimo* foram usados dois *arrays* (`a` e `pos_a`). Com que objetivo?

N.º ______ Nome ______

9. [1.0] Complete: "Dados n pontos no plano, o algoritmo Graham-scan calcula ______________________ ______________________________________. Tem complexidade ____________ se se usar um algoritmo de ordenação com complexidade ____________ no **pior caso**, como ______________________________ ou ______________________________".

10. 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) [0.5] Indique T_{kn}, S_{kn} e N_{kn}, para $0 \leq n \leq 5$ e $1 \leq k \leq 3$, sendo R dada por:

```
15 35 45 60 65
25 50 55 55 55
20 30 55 60 60
```

b) [1.0] Apresente a recorrência que define T_{kn}, S_{kn} e N_{kn}, para $k \geq 1$ e $n \geq 0$.

c) [1.5] Adaptando a função dada nas aulas, escreva (em pseudocódigo) a função 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.

<u>Resolva esta alínea no verso da folha.</u>

(Fim)