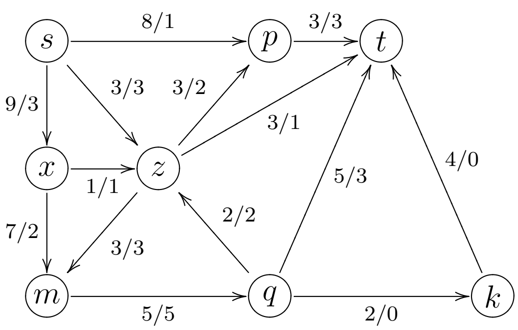
Desenho e Análise de Algoritmos (CC2001) 2017/18

Exame (29.01.2018) *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:

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

b) [1.5] Partindo do fluxo 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 os passos).

c) [0.6] Complete as frases: *A capacidade do corte* $(\{s,q,t\},\{p,x,z,m,k\})$ *é* ____ .

____ *é um corte* $\{S,T\}$ *com capacidade mínima, a qual é* ____ .

d) [1.2] Escreva em pseudocódigo uma **função** que determine **um caminho para aumento** numa dada rede residual G_f e o **incremento** de $|f|$. Assuma que $V = \{1, 2, \ldots, |V|\}$. Qual é a sua complexidade?

2. [1.0] Considere o problema de formar uma certa quantia q com um número **mínimo** de moedas de valores 1, 2, 5, 10, 20, 50, 100, e 200, estando q e esses valores na mesma unidade monetária. Indique a **estratégia *greedy*** que obtém a solução ótima se se dispuser de um número ilimitado de moedas de cada tipo e prove que é incorreta se **for limitado**. Indique **todos** os erros possíveis e instâncias nessas condições.

3. [2.0] Usando a definição matemática das ordens de grandeza e das classes indicadas, justifique a veracidade ou falsidade de cada uma das afirmações seguintes.

a) $3n^2 + 100 \in \Omega(6n^2 + 5)$.

(Continua, v.p.f.)

N.º | | Nome | |

b) $100n + 3n\log_2 n \notin \Theta(n \log_2 n)$

4. Considere o algoritmo de Dijkstra, suportado por uma *heap binária de mínimo* Q, para determinação de caminhos mínimos com origem num nó s num grafo dirigido $G = (V, E, d)$, com $d(e) \in \mathbb{Z}^+$, para $e \in E$.

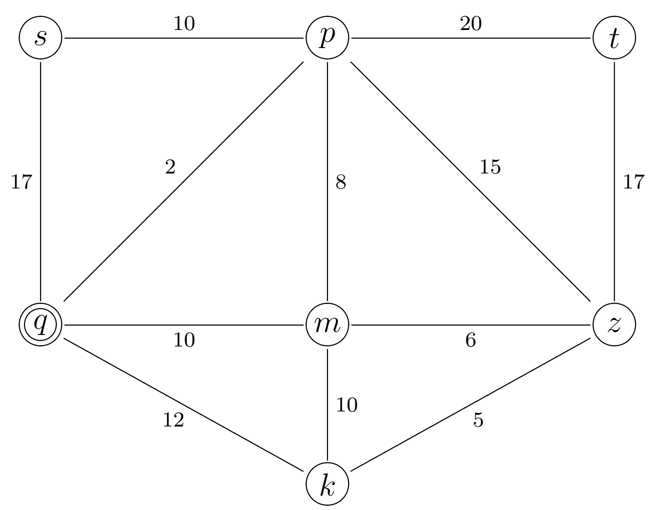
a) [0.3+0.7] O que retorna a operação $\textsc{ExtractMin}(Q)$? Como é efetuada e de que modo afeta Q na implementação de Q apresentada nas aulas (recorde que se mantém dois *arrays* $Q.a$ e $Q.pos\_a$).

b) [1.0] Indique a complexidade de $\textsc{ExtractMin}(Q)$ ______, $\textsc{DecreaseKey}(Q, v, dist[v])$ ______, e do algoritmo de Dijkstra ______.

5. [0.4] Uma árvore de pesquisa *red-black* não é uma árvore equilibrada. Que propriedade garante que a operação de procura de um dado valor seja realizada em $O(\log_2 n)$, sendo n o número de valores na árvore?

/* Em alternativa, resolva questão 10. */

6. [2.0] Aplique o algoritmo de Prim para obter uma árvore geradora $\mathcal{T}$ de peso **mínimo** do grafo indicado, com raiz q. Anote os nós com pares $(dist, pai)$, como se definiu nas aulas, de modo a poder reconstruir **os passos intermédios** dessa aplicação. Na caixa à direita, indique os nós em $\mathcal{T}$ após cada iteração.



7. Considere o algoritmo de Kosaraju-Sharir para determinação das componentes fortemente conexas de um grafo dirigido $G = (V, E)$. Pretendemos obter uma lista de listas de nós que definem cada componente.

a) [1.5] Descreva os passos principais, as suas complexidades temporais e as estruturas de dados que usam.

b) [0.4] Que propriedades da pesquisa e do grafo de componentes são determinantes para a correção?

N.º [ ] Nome [ ]

8. Considere a função ANALISAROTA(s, t, m, L) para verificar se uma rota a dar por um utilizador passa em s e t e tem lugares suficientes entre s e t para um grupo de m elementos, sendo L uma matriz e $L[v, w]$ o número de lugares disponíveis no troço (v, w) (que será -1 se não existir esse troço). A rota é dada pela sequência de nós por onde passa, os quais são todos distintos. O utilizador começa por dar o número de nós da rota e a seguir indicará os nós. Assuma que $s \neq t$ e $m \geq 1$.

```
ANALISAROTA(s, t, m, L)
  d ← m;
  ok ← false;
  ler(n);  ler(v);   k ← 1;
  Se (v = s) então ok ← true;
  Enquanto (v ≠ t ∧ d = m ∧ k < n) fazer
      ler(w);   k ← k + 1;
      Se (w = s) então ok ← true;
      senão
          se (ok = true ∧ d > L[v, w]) então
              d ← L[v, w];
      v ← w;
  Se (v = t ∧ ok = true ∧ d = m) então
      retorna true;
  retorna false;
```

a) [1.2] Qual é a complexidade no pior caso? [ ]

E, no melhor caso? [ ] Identifique-os e explique.

b) [0.7] Assuma que não é necessário ler a rota até ao fim. Indique um **invariante de ciclo** que permita demonstrar a correção da função.

c) [1.1] Usando **indução matemática** (sobre o número de vezes que testa a condição de ciclo), **demonstre o invariante** que indicou e, **aplicando-o**, apresente a dedução de que a função retorna o valor correto.

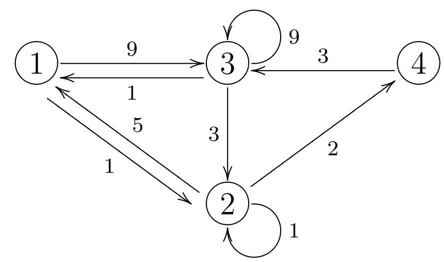
d) [0.3] Assuma que é necessário ler a rota até ao fim. Corrija o programa.

(Continua, v.p.f.)

N.º [ ] Nome [ ]

9. Seja $G = (V, E, d)$ um grafo dirigido finito, com $V = \{1, 2, \ldots, n\}$, e em que $d(e) \in \mathbb{Z}^+$ define o peso do ramo e, para todo $e \in E$. O peso de um percurso é a soma dos pesos nos ramos do percurso. Considere percursos $\gamma_{ij}^{(k.r)}$ de i para j com no máximo r ramos e que passam num nó k pré-definido (basta que k ocorra, não tem de ser um nó intermédio). Seja $K_{ij}^{(k,r)}$ o **peso mínimo** que um tal percurso pode ter, para k e r fixos. Um percurso tem pelo menos um ramo. Se o percurso não existir, defina $K_{ij}^{(k,r)}$ como ∞.

a) [0.6] Para a instância representada, indique os valores de:



$K_{3,3}^{(3,2)}$ [ ] $K_{1,4}^{(3,2)}$ [ ] $K_{1,4}^{(3,9)}$ [ ]

$K_{1,1}^{(3,6)}$ [ ] $K_{4,4}^{(3,2)}$ [ ] $K_{2,2}^{(3,4)}$ [ ]

b) [0.5] No caso geral, prove que nenhum percurso $\gamma_{ij}^{(k,r)}$ com peso $K_{ij}^{(k,r)}$ contém um ciclo de k para k, a menos que $i = j = k$.

c) [1.0] Defina $K_{ij}^{(k,r)}$ por uma recorrência, para todo $(i, j) \in V \times V$ e $r \geq 1$, com $k \geq 1$ fixo, bem como o nó $N_{ij}^{(k,r)}$ que segue o nó i num percurso $\gamma_{ij}^{(k,r)}$ encontrado com peso $K_{ij}^{(k,r)}$. **Explique sucintamente.** (*Sugestão:* como exprimir a matriz $K^{(k,r+1)}$ a partir da matriz $K^{(k,r)}$ e de d?)

d) [0.2] Indique um valor r_0, dependente de n, tal que $K_{ij}^{(k,r)} = K_{ij}^{(k,r_0)}$, para todo (i,j) e $r \geq r_0$.

e) [1.1] Escreva (em pseudocódigo) uma função RESOLVE(D, n, k, K), com **complexidade** $O(n^3)$, para obter a matriz K, sendo K_{ij} o peso mínimo de um percurso de i para j que passe por k, com $k \geq 1$ fixo, para todos os pares (i,j). Deve ser baseada na recorrência definida anteriormente e usar **programação dinâmica**. São dados n, k e a matriz D, sendo $D_{ij} = d(i,j)$ se $(i,j) \in E$ (caso contrário, $D_{ij} = \infty$).

10. [0.4] Explique de que modo a correção do algoritmo de Kruskal, para cálculo de uma árvore de suporte de peso **máximo** (ou **mínimo**) de um grafo $G = (V, E, d)$, se deduz da correção da estratégia *greedy* que determina um conjunto máximo independente num matróide pesado $(S, \mathcal{F})$. A que corresponde S e $\mathcal{F}$?

/* Em alternativa, resolva questão 5. */

(Fim)