INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

Ano Lectivo 2017/2018 — Exame - versão A

05 de julho de 2018 — Duração: 3h00m

- O tempo mínimo de permanencia na sala é 1 hora. Não serão permitidas entradas após 30 minutos do início da prova.
- Só serão avaliadas as respostas **legíveis** apresentadas nas grelhas desta página e da seguinte, com a excepção das perguntas com $\langle$ XXX $\rangle$, que serão corrigidas na folha do enunciado respectiva e será avaliado o desenvolvimento da resposta.
- Nas questões de verdadeiro, deconhecido ou falso, cada resposta errada desconta meia certa.
- Certifique-se que a sua identificação está legível nas folhas com as suas soluções.

I. (1,25 + 1,25 + 1,25 + 1,25 + 1,25 + 1,25= 7,5 val.)

I.a)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$rank[x_i]$										
$p[x_i]$										

I.b)

	A	B	C	D	E	F	G	H	I	J
d/low	/	/	/	/	/	/	/	/	/	/

$SSCs$:				

I.c)

Ordem arcos								
Custo MST								
Nº MST								

I.d)

	A	B	C	D	E	F	G	H
$h()$								

$\widehat{w}(A,B)$	$\widehat{w}(B,E)$	$\widehat{w}(C,E)$	$\widehat{w}(E,G)$	$\widehat{w}(G,H)$

I.e)

Expressão	
Majorante	

I.f)

	A	B	C	D
$h()$				
Corte :	/		$f(S,T) =$	

II. (1,0 + 1,5 = 2,5 val.)

II.a) $\langle$ XXX $\rangle$

II.b) $\langle$ XXX $\rangle$

Número:________ Nome:________

III. (1,25 + 1,5 + 1,0 = 3,75 val.)

III.a)

	$\pi[4]$	$\pi[6]$	$\pi[8]$	$\pi[9]$	$\pi[10]$
Valor					

III.b)

Primeira	$Z =$	$x_1 =$	$x_2 =$	$x_3 =$
Ótima	$Z =$	$x_1 =$	$x_2 =$	$x_3 =$
Dual Ótima	$Z' =$		$y_1 =$	$y_2 =$

III.c)

	a	b	c	d	e	f
Codificação						

Total Bits	

IV. (1,0 + 1,25 = 2,25 val.)

IV.a)

maior	número de moedas	Solução

IV.b) $\langle$ XXX $\rangle$

V. (1,0 + 1,0 = 2,0 val.)

V.a)

Ordem	

V.b)

$\delta(2,a)$	$\delta(5,b)$	$\delta(6,a)$	$\delta(9,a)$	$\delta(10,a)$

VI. (1,0 + 1,0 = 2,0 val.)

VI.a)

	a)	b)	c)	d)	e)
Resposta					

VI.b) $\langle$ XXX $\rangle$

I. (1,25 + 1,25 + 1,25 + 1,25 + 1,25 + 1,25= 7,5 val.)

I.a) Considere o seguinte conjunto de operações sobre conjuntos disjuntos:

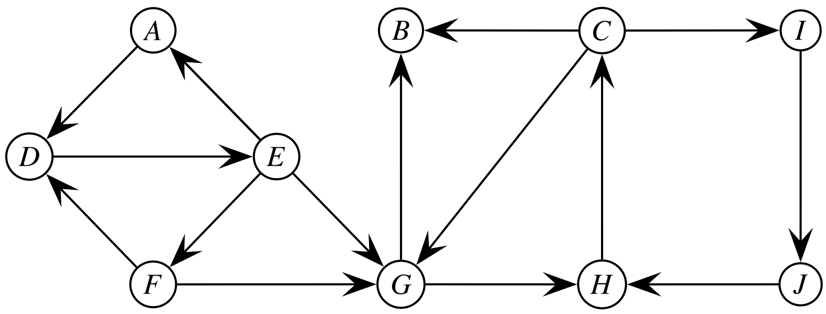
---

for i = 1 to 10 Make-Set (x_i)
for i = 1 to 5 Union (x_{2i}, x_{2i-1})
Union (x_1, x_{10})
j = Find-Set(x_2)
k = Find-Set(x_7)
Union (x_7, x_4)
Union (j, x_8)
Union (x_6, k)

Use a estrutura em árvore para representação de conjuntos disjuntos com a aplicação das heurísticas de união por categoria e compressão de caminhos. Para cada elemento x_i, indique os valores de categoria ($rank[x_i]$) e o valor do seu pai na árvore ($p[x_i]$).

Nota: Na operação Make-Set(x), o valor da categoria de x é inicializado a 0. Na operação de Union(x, y), em caso de empate, considere que o representante de y é que fica na raíz.

I.b) Considere o grafo dirigido:



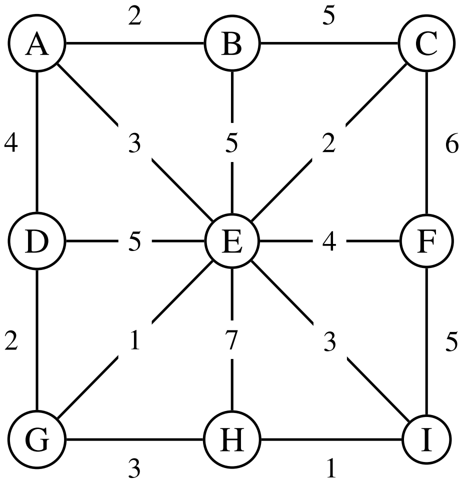
Aplique o algoritmo de Tarjan para encontrar os componentes fortemente ligados do grafo. Os vértices são sempre considerados por ordem lexicográfica (ou seja, A, B, C...). Os adjacentes também são sempre considerados por ordem lexicográfica.

Para cada vértice indique os valores de descoberta d e low após a aplicação do algoritmo.

Indique os componentes fortemente ligados **pela ordem que são descobertos pelo algoritmo**.

Nota: Neste algoritmo os valores de d começam em 1.

I.c) Considere o grafo não dirigido e pesado da figura.

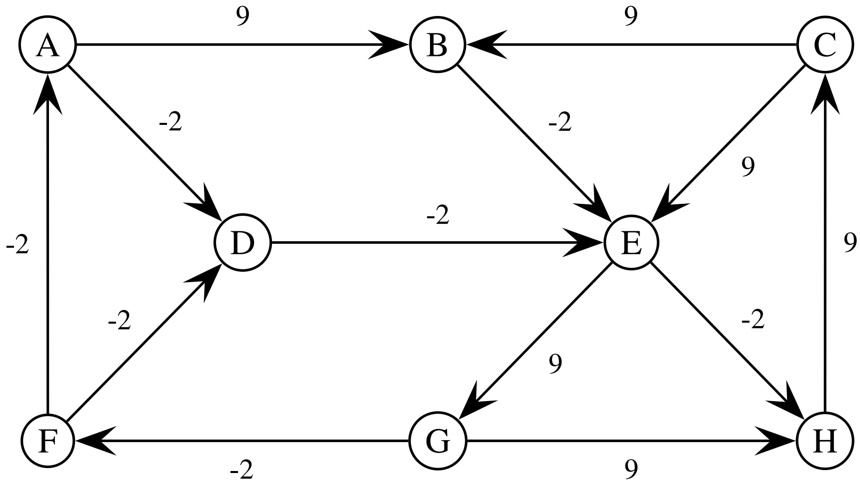


Aplique o algoritmo de Kruskal ao grafo.

Indique a ordem pela qual os arcos são seleccionados para pertencer à árvore abrangente de menor custo (MST). Em caso de empate, considere os vértices por ordem lexicográfica.

Indique o custo de uma MST e quantas MST diferentes existem no grafo.

I.d) Considere o grafo dirigido e pesado da figura.



Considere a aplicação do algoritmo de Johnson ao grafo. Calcule os valores de $h(u)$ para todos os vértices $u \in V$ do grafo.

Indique, os pesos dos seguintes arcos após a repesagem: (A,B), (B,E), (C,E), (E,G), (G,H).

I.e) Considere a função recursiva:

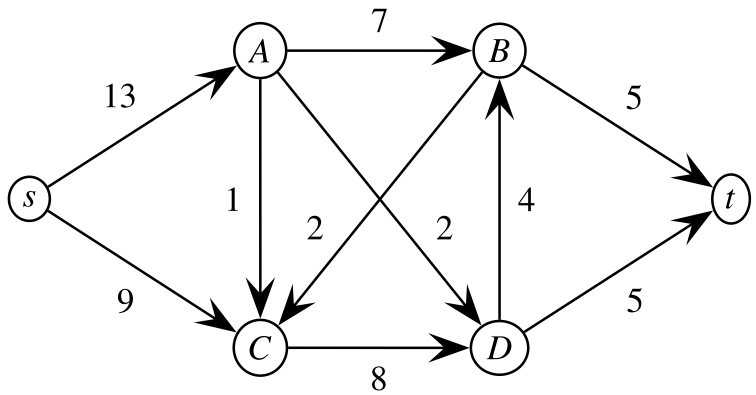
```
int f(int n) {
  int i = n*n, j = 0;

  while(i > 0) {
    i = i / 2;
    j++;
  }

  if(n > 1)
    i = i * f(n/2) + f(n/2);

  while (j > 0) {
    i++;
    j--;
  }
  return i;
}
```

Indique a expressão (recursiva) que descreve o tempo de execução da função em termos do número n, e de seguida, utilizando os métodos que conhece, determine o menor majorante assimptótico.

I.f) Considere a rede de fluxo da figura onde s e t são respectivamente os vértices fonte e destino na rede.

Aplique o algoritmo Relabel-To-Front na rede de fluxo. Considere que a lista de vértices é inicializada por ordem alfabética e que os vizinhos de cada vértice também estão ordenados alfabeticamente. Assim, as listas de vizinhos dos vértices intermédios são as seguintes:

$N[A] = <B,C,D,s>$ $N[B] = <A,C,D,t>$ $N[C] = <A,B,D,s>$ $N[D] = <A,B,C,t>$



Indique a altura final de cada vértice. Indique ainda o corte mínimo da rede e o valor do fluxo máximo.

II. (1,0 + 1,5 = 2,5 val.)

II.a) Considere uma rede de fluxo $G = (V, E)$. Suponha que foi calculada uma função de fluxo $f : E \to N$ que define o fluxo máximo na rede de fluxo G.

Considere que é adicionado um novo arco (u, v) à rede de fluxo com capacidade k. Ou seja, temos uma nova rede $G' = (V, E \cup \{(u, v)\})$ onde $u, v \in V$.

Proponha um algoritmo que calcule o fluxo máximo da rede G', actualizando a função de fluxo f calculada anteriormente para G.

Indique a complexidade do algoritmo proposto.

II.b) Considere um grafo $G = (V, E)$ dirigido e sem pesos. Um vértice $u \in V$ é denominado como vértice raíz se para todos os outros vértices $v \in V$ existe um caminho de u para v. Ou seja, todos os vértices do grafo são atingíveis a partir de um vértice raíz.

Defina um algoritmo eficiente que permite identificar **todos** os vértices raíz de um grafo G. Note que o grafo G pode não ter um vértice raíz, pelo que o seu algoritmo deve identificar essa situação.

Indique a complexidade da solução proposta.

III. (1,25 + 1,5 + 1,0 = 3,75 val.)

III.a) Considere o algoritmo de Knuth-Morris-Pratt. Dado o padrão $P = \texttt{aabaaabaab}$, calcule a função de prefixo $\pi[k]$ para o padrão P. Indique os valores de $\pi[4]$, $\pi[6]$, $\pi[8]$, $\pi[9]$ e $\pi[10]$.

III.b) Considere o seguinte programa linear:

$$
\begin{array}{lrcl}
\text{Maximizar} & -4x_1+5x_2-4x_3 & & \\
\text{Sujeito a} & -\ x_1+\ x_2+\ x_3 & \leq & -2 \\
 & -3x_1+4x_2+3x_3 & \leq & 1 \\
 & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

Indique o valor da função objectivo e o respectivo valor das variáveis x_1, x_2 e x_3 na **primeira solução exequível** encontrada pelo algoritmo Simplex. Em caso de empate em algum critério de aplicação do algoritmo, aplique a regra de Bland. Ou seja, escolha a variável de menor índice.

Indique também o valor da função objectivo e o respectivo valor das variáveis para a solução ótima e para a solução do sistema dual, com a variável y_1 associada à primeira restrição e a variável y_2 associada à segunda restrição.

III.c) Considere o problema de compressão de dados de um ficheiro usando a codificação de Huffman. Indique o código livre de prefixo ótimo para cada caractere num ficheiro com 100 caracteres com o seguinte número de ocorrências: $f(a) = 1, f(b) = 2, f(c) = 5, f(d) = 13, f(e) = 34, f(f) = 45$. Quando constrói a árvore, considere o bit 0 para o nó com menor frequência.

Indique também o total de bits no ficheiro codificado.

IV. (1,0 + 1,25 = 2,25 val.)

IV.a) Considere o problema dos trocos utilizando programação dinâmica. Assuma que apenas existem moedas com dois valores, a moeda d_1 com valor 4 e a moeda d_2 com valor 5. É possivel utilizar várias moedas com o mesmo valor, por exemplo para fazer um troco de 16 é possível utilizar quatro moedas d_1. Existem quantias que não podem ser obtidas com estas moedas, por exemplo não é possível obter um troco de valor 3.

Utilizando programação dinâmica identifique o maior troco que não pode ser obtido com estas moedas. Indentifique também qual o menor número de moedas necessárias para obter um troco de valor 17 e quais são as moedas necessárias.

IV.b) Nesta questão iremos utilizar um algoritmo de programação dinâmica para encontrar uma sub-sequência alternada que maximiza a soma total. Os dados consistem numa sequência de números não negativos S que podem ser referênciados como S_i para algum índice i a começar em 1. Por exemplo, para a sequência $S = (5,2,1,7)$, temos $S_2 = 2$.

O objectivo é obter uma sub-sequência com a maior soma possível, respeitando a regra que a sub-sequência não pode ter índices consecutivos, por exemplo se $S_1 = 5$ for selecionado então $S_2 = 2$ não pode fazer parte da sub-sequência. Neste caso a solucão seria $S_1 + S_4 = 5 + 7 = 12$.

Utilizaremos um vector `D[i]` em que `i` varia entre 0 e o tamanho de S. Cada valor `D[i]` deverá indicar a maior soma que pode ser obtida a partir de uma sub-sequência alternada S' dos primeiros i elementos de S.

Complete a fórmula da recursão para a resolução deste problema.

$$\texttt{D[i]} = \begin{cases} \boxed{} & \text{, se } i = 0 \\ \boxed{} & \text{, se } i = 1 \\ \boxed{} & \text{, se } i > 1 \end{cases}$$

V. (1,0 + 1,0 = 2,0 val.)

V.a) Considere o seguinte conjunto de clientes que tem tarefas para ser processadas por um servidor. Cada tarefa de demora s_i unidades de tempo a processar. O objetivo é minimizar o tempo total de espera dos clientes, indique a ordem porque devem ser processadas as tarefas para atingir este objectivo.

i	1	2	3	4	5	6	7	8
s_i	10	11	5	9	7	15	13	12

V.b) Considere o autómato finito determinista para o padrão $P = aabaaabaab$, indique os seguintes valores de transição: $\delta(2,a)$, $\delta(5,b)$, $\delta(6,a)$, $\delta(9,a)$, $\delta(10,a)$.

VI. (1,0 + 1,0 = 2,0 val.)

VI.a) Suponha que o Prof. Caracol descobriu um algoritmo polinomial para resolver o problema 3-COLORING. Considerando esta descoberta do Prof. Caracol, para cada uma das afirmações seguintes, indique se é verdadeira (**V**), falsa (**F**) ou se não se sabe (**D**).

a. 3-COLORING $\in$ NP
b. P = NP
c. CLIQUE $\notin$ NP-HARD
d. 2-COLORING $\in$ P
e. 3-CNFSAT $\notin$ NP-HARD

VI.b) O problema **2-CLIQUE** de um determinado grafo não dirigido G_2 com um conjunto vértices V e arcos E consiste em encontrar dois cliques C_1 e C_2 disjuntos e possivelmente vazios que no total contenham pelo menos k_2 vértices. As seguintes propriedades definem matematicamente o **2-CLIQUE**:

$$
\begin{aligned}
&C_1 \subseteq V \\
&C_1 \times C_1 \subseteq E \\
&C_2 \subseteq V \\
&C_2 \times C_2 \subseteq E \\
&C_1 \cap C_2 = \emptyset \\
&|C_1 \cup C_2| \geq k_2
\end{aligned}
$$

Recorde o problema **CLIQUE** que encontra um clique num grafo não dirigido G_1. Sabendo que o problema **CLIQUE** é NP-Completo, prove que o problema **2-CLIQUE** é NP-Completo. Prove primeiro que **2-CLIQUE** $\in$ NP.