INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

Ano Lectivo 2017/2018 — Repescagem 1º Teste - versão B

05 de Julho de 2018 — Duração: 1h30m

---

- **A prova é sem consulta**.
- O tempo mínimo de permanência 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, com a excepção das perguntas com $\underline{\langle \text{ XXX } \rangle}$, que serão corrigidas na folha do enunciado respectiva e será avaliado o desenvolvimento da resposta.
- Certifique-se que a sua identificação está legível nas folhas de resposta.

---

**I. (2,5 + 2,5 + 2,5 + 2,5 + 2,5 + 2,5 = 15,0 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 |
|---|---|---|---|---|---|---|---|---|
| $h()$ | | | | | | | | |

| $\widehat{w}(A,B)$ | $\widehat{w}(B,H)$ | $\widehat{w}(C,H)$ | $\widehat{w}(E,F)$ | $\widehat{w}(H,E)$ |
|---|---|---|---|---|
| | | | | |

**I.e)**

| | |
|---|---|
| Expressão | |
| Majorante | |

**I.f)**

| | A | B | C | D |
|---|---|---|---|---|
| $h()$ | | | | |

| | | | |
|---|---|---|---|
| Corte : | / | $f(S,T) =$ | |

**II. (2,0 + 3,0 = 5,0 val.)**

**II.a)** $\underline{\langle \text{ XXX } \rangle}$ **II.b)** $\underline{\langle \text{ XXX } \rangle}$

Número:________ Nome:________________ 

**I. (2,5 + 2,5 + 2,5 + 2,5 + 2,5 + 2,5 = 15,0 val.)**

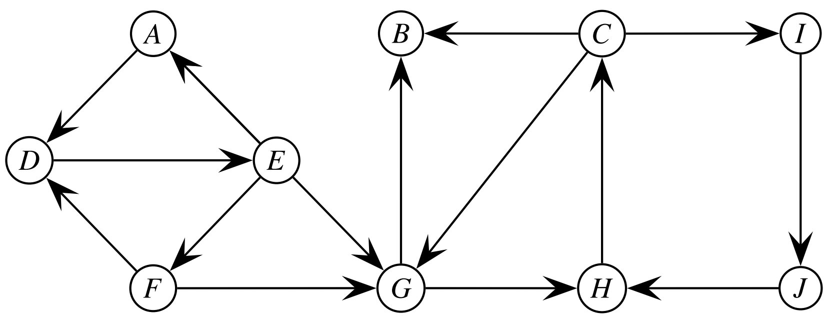
**I.a)** Considere o seguinte conjunto de operações sobre conjuntos disjuntos:

**for** $i$ = *1 to 10* **do**<br>
  Make-Set ($x_i$)<br>
**for** $i$ = *1 to 5* **do**<br>
  Union ($x_{2i-1}, x_{2i}$)<br>
Union ($x_2, x_9$)<br>
j = Find-Set($x_1$)<br>
k = Find-Set($x_7$)<br>
Union ($x_7, x_4$)<br>
Union (j, $x_8$)<br>
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:



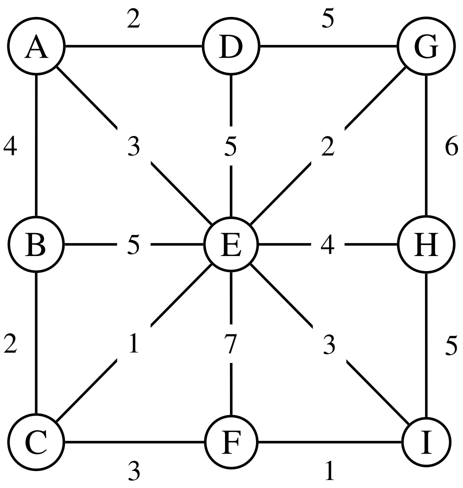

Aplique o algoritmo de Tarjan para encontrar os componentes fortemente ligados do grafo. Os vértices são sempre considerados por ordem lexicográfica (ou seja, A, B, C...). Os adjacentes também são sempre considerados por ordem lexicográfica.

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.

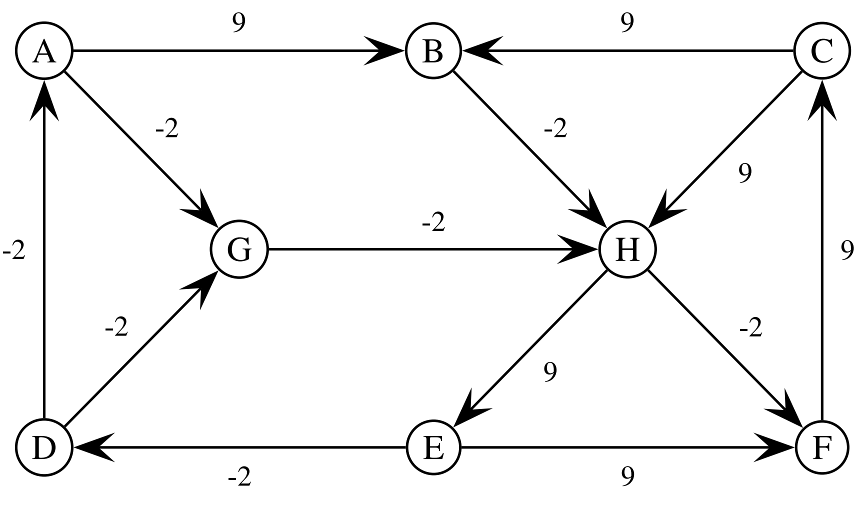
**I.c)** Considere o grafo não dirigido e pesado da figura.




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: $(A,B)$, $(B,H)$, $(C,H)$, $(E,F)$, $(H,E)$.

**I.e)** Considere a função recursiva:

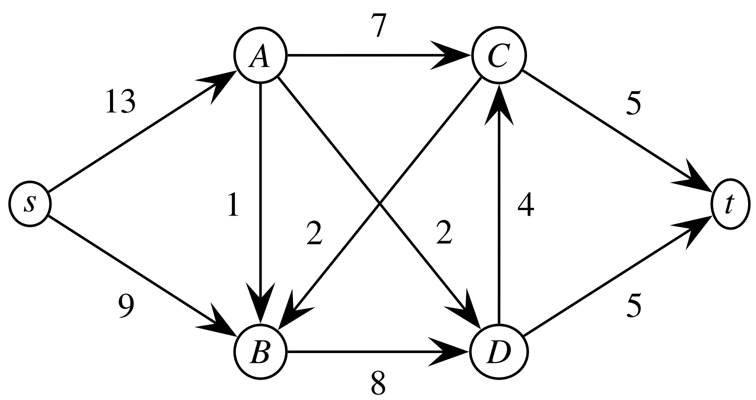
```
int f(int n) {
  int i = n*n, j = 0;

  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:

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




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

**II. (2,0 + 3,0 = 5,0 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.