Instituto Superior Técnico

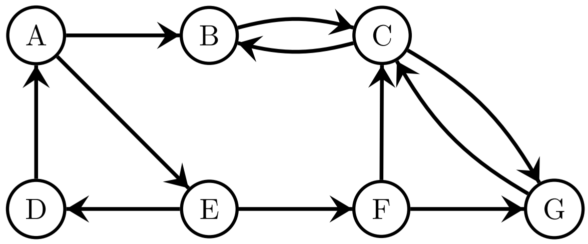
Análise e Síntese de Algoritmos

Ano Lectivo 2019/2020 Repescagem 1º Teste

RESOLUÇÃO

I. (2,5 + 2,5 + 2,5 + 2,5 = 10,0 val.)

I.a)

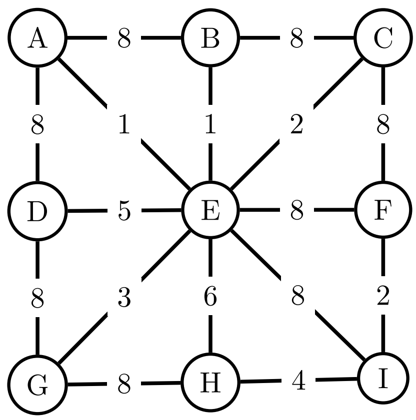


Considere o grafo dirigido da figura. Aplique o algoritmo de Tarjan para identificar os componentes fortemente ligados, considerando o vértice **D** como inicial. Durante a aplicação do algoritmo considere que tanto a escolha dos vértices a visitar, como a pesquisa dos vértices adjacentes são feitas por ordem lexicográfica (ou seja, A, B, C, ...).

Indique os componentes fortemente ligados que são identificados pelo algoritmo e o valor *low* calculado para cada vértice. Considere que o tempo de descoberta d começa em 1.

	A	B	C	D	E	F	G
low()							
SCCs :							

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



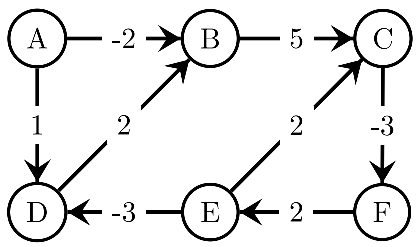
Considere a execução do algoritmo de Kruskal para determinar árvores abrangentes de menor custo, até processar arcos de peso 4, inclusivé.
Indique a soma do peso dos arcos seguros seleccionados, bem como quais os conjuntos disjuntos existentes nessa fase do algoritmo.

Soma pesos:	
Conjuntos disjuntos:	

Indique ainda o custo da MST calculada.

Custo MST:	

I.c) Considere a aplicação do algoritmo de Johnson ao grafo dirigido e pesado da figura.

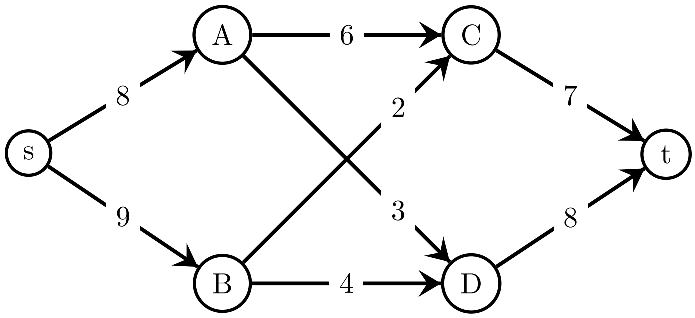


Calcule os valores de $h(u)$ para todos os vértices $u \in V$ do grafo. Calcule também os pesos de todos os arcos após a repesagem.

	A	B	C	D	E	F
$h()$						

$\widehat{w}(A, B)$	$\widehat{w}(A, D)$	$\widehat{w}(B, C)$	$\widehat{w}(C, F)$	$\widehat{w}(D, B)$	$\widehat{w}(E, C)$	$\widehat{w}(E, D)$	$\widehat{w}(F, E)$

I.d) Aplique o algoritmo de Edmonds-Karp na seguinte rede de fluxo, onde s e t são respectivamente os vértices fonte e destino na rede.



Indique um corte mínimo da rede, o valor do fluxo máximo, e o fluxo de cada arco após a aplicação do algoritmo. Nota: Na selecção do caminho de aumento, em caso de empate (caminhos de aumento com o mesmo comprimento), escolha o menor caminho de aumento por ordem lexicográfica.

f(s,A)	f(s,B)	f(A,C)	f(A,D)	f(B,C)	f(B,D)	f(C,t)	f(D,t)

Corte :	/	f(S,T) =	

II. (2,5 + 2,5 + 2,5 + 2,5 = 10 val.)

II.a) Considere a função recursiva:

```
int f(int n) {
   int x = 0;

   for (int i = 0; i < n; i++) { // Loop 1
      for (int j=0; j < i; j++) { // Loop 2
         x++;
      }
   }

   if ((n > 0) && ((n%2) == 1)) {
      x = x + f(n - 1);
   }
   else if ((n > 0) && ((n%2) == 0)) {
      x = 2*f(n/2);
   }

   return x;
}
```

1. Determine um upper bound medido em função do parâmetro n para o número de iterações dos loops **1** e **2** por cada chamada à função f.

2. Determine o menor majorante assimptótico da função f em termos do número n utilizando os métodos que conhece.

Solução:

1. Upper bounds:

    - **Loop 1:** $\sum_{i=0}^{n-1} 1 = n \in O(n)$
    - **Loop 2:** $\sum_{i=0}^{n-1} \left(\sum_{j=1}^{i} 1 \right) = \sum_{i=0}^{n-1} i = \sum_{i=1}^{n-1} i = \frac{n*(n+1)}{2} \in O(n^2)$

2. Analisamos separadamente os casos n é par e n é ímpar.

    - n é par: $T(n) = T(n/2) + O(n^2)$
    - n é ímpar: $T(n) = T(n-1) + O(n^2) = T((n-1)/2) + O(n^2) + O((n-1)^2) \leq T(n/2) + O(n^2)$

    Aplicando o Teorema Mestre ($a = 1$, $b = 2$, $d = 2$) concluímos que: $T(n) \in O(n^2)$.

II.b) O geógrafo João Caracol foi encarregado de fazer o levantamento das ilhas de Caracolândia com base num mapa pré-calculado. O mapa consiste numa matriz de píxeis, sendo que uma ilha corresponde a um conjunto de píxeis = 1 ligados entre si. Dizemos que um píxel está ligado a outro se estiver ligado horizontal, vertical ou diagonalmente. Por exemplo, a seguinte matriz:

$$\begin{array}{cccccccc}
\mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\
\mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} \\
\mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 \\
\mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0
\end{array}$$

contém três ilhas. Proponha um algoritmo eficiente para determinar o número de ilhas num dado mapa e indique a respectiva complexidade assimptótica. *Nota:* A complexidade do algoritmo proposto deve ser apresentada em função das dimensões da matriz dada como input ($n \times m$).

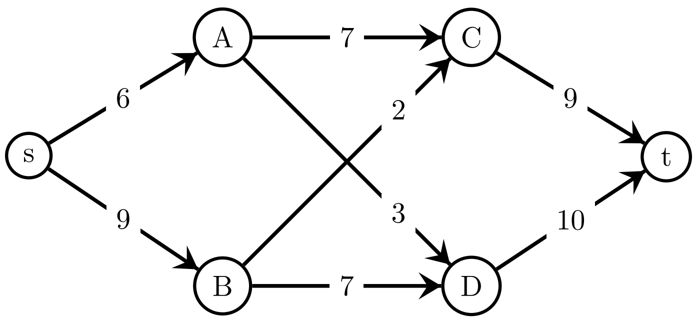
Solução:

Descrição da solução: O João Caracol deve construir um grafo a partir da matriz, com um vértice para cada posição da matriz e um arco entre cada dois vértices correspondentes a posições adjacentes da matriz. Deve depois executar uma DFS no grafo obtido começando sempre a operação DFS-Visit num vértice correspondente a uma posição com um **1** na matriz. Para tal, durante a construcão do grafo, o João Caracol deve guardar os vértices correspondentes a posições com um **1** numa lista ligada.

Análise da complexidade:

- $|V| = n.m \in O(n.m)$
- $|E| \leq n.m.8 \in O(n.m)$
- Construção do grafo: $O(n.m)$
- DFS: $O(V + E) = O(n.m)$
- Complexidade total: $O(n.m)$

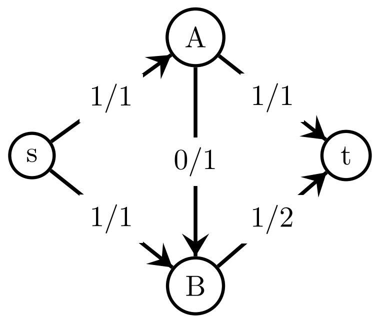
II.c) Numa rede de fluxo com capacidades inteiras, um arco é considerado um *crucial* se, diminuindo o valor da sua capacidade em uma unidade, o valor do fluxo máximo também diminui. Considere a seguinte rede de fluxo:



1. Indique os arcos cruciais da rede fluxo apresentada em cima.
2. Dê um exemplo de uma rede de fluxo G, com no máximo 4 vértices, e de um fluxo máximo f^* em G, tais que nem todos os arcos saturados de G são arcos cruciais.
3. Dado um fluxo máximo f^* numa rede de fluxo G e a respectiva rede residual G_{f^*}, proponha um algoritmo eficiente para determinar se um dado arco (u, v) de G é crucial e identifique a respectiva complexidade.

Solução:

1. Arcos cruciais: $\{(s, A), (s, B), (B, D), (B, C)\}$
2. A rede de fluxo é apresentada em baixo:



3. Dado um arco (u, v) e a rede residual G_{f^*}, (u, v) é crucial se a sua capacidade residual for 0 e se:
   - Não for possível atingir o vértice t a partir de u em G_{f^*} sem usar o arco (u, v); e
   - Não for possível atingir o vértice v a partir de s em G_{f^*} sem usar o arco (u, v).

   Basta, portanto, efectuar duas DFS's em G_{f^*}:

   - DFS a partir do vértice u sem considerar o arco (u, v) e verificar se é possível chegar a t;
   - DFS a partir do vértice s sem considerar o arco (u, v) e verificar se é possível chegar a v.

   Complexidade: $O(V + E)$.

II.d) O Departamento de Informática da Universidade Técnica de Caracolândia decidiu implementar uma nova aplicação para determinar automaticamente a composição dos júris de dissertações de mestrado. No semestre em consideração existem n estudantes que pretendem defender as suas dissertações, $\{S_1, ..., S_n\}$, e m professores disponíveis para participar em júris, $\{P_1, ..., P_m\}$. O problema da constituição dos júris deve respeitar as seguintes restrições:

- Cada professor P_j, com $1 \leq j \leq m$, pode participar em no máximo l_j júris;
- Cada júri deve ser constituído por três professores.

Admita que o problema tem solução, isto é, que, dadas as disponibilidades dos professores, é possível constituir o júri de todos os alunos. Pretende-se agora calcular uma atribuição de professores a júris.

1. Modele o problema da constituição de júris como um problema de fluxo máximo. A resposta deve incluir o procedimento utilizado para determinar a constituição dos júris a partir do fluxo calculado.

2. Indique o algoritmo que utilizaria para a calcular o fluxo máximo, bem como a respectiva complexidade assimptótica medida em função dos parâmetros do problema (número de alunos, n, e número de professores, m).
   Nota: De entre os algoritmos de fluxo estudados nas aulas deve escolher aquele que garanta a complexidade assimptótica mais baixa para o problema em questão.

Solução:

1. *Construção da rede de fluxo $G = (V, E, w, s, t)$.* Na construção da rede de fluxo consideramos um vértice por professor, um vértice por estudantes e e dois vértices adicionais s e t, respectivamente a fonte e o sumidouro. Formalmente:

   - $V = \{S_i \mid 1 \leq i \leq n\} \cup \{P_j \mid 1 \leq i \leq m\} \cup \{s, t\}$
   -
   $$\begin{array}{ll} E = \{(s, P_j, l_j) \mid 1 \leq j \leq m\} & P_j \text{ só pode participar em } l_j \text{ defesas} \\ \quad \cup \; \{(P_j, S_i, 1) \mid j \in \mathtt{SP}(i)\} & P_j \text{ pode participar no júri do estudante } S_i \\ \quad \cup \; \{(S_i, t, 3) \mid 1 \leq i \leq n\} & 3 \text{ membros por júri} \end{array}$$

   Como sabemos que é possível formar todos os jurís, concluímos que o fluxo máximo é $3.n$. Uma vez calculado o fluxo máximo, f^*, o júri do aluno S_i é constituído pelos professores P_{j_1}, P_{j_2} e P_{j_3} tais que: $f^*(P_{j_1}, S_i) = 1$, $f^*(P_{j_2}, S_i) = 1$, e $f^*(P_{j_3}, S_i) = 1$.

2. *Complexidade.*

   - $|V| = n + m + 2 \in O(n + m)$
   - $|E| = m + m.n + n \in O(n.m)$
   - $|f^*| \leq 3n \in O(n)$
   - Ford-Fulkerson: $O(n^2.m)$
   - RF: $O((n + m)^3$

   A melhor solução consiste em usar um algoritmo baseado no método de Ford Fulkerson.