1º Teste de Análise e Desenho de Algoritmos

Duração: 1 hora e 45 minutos

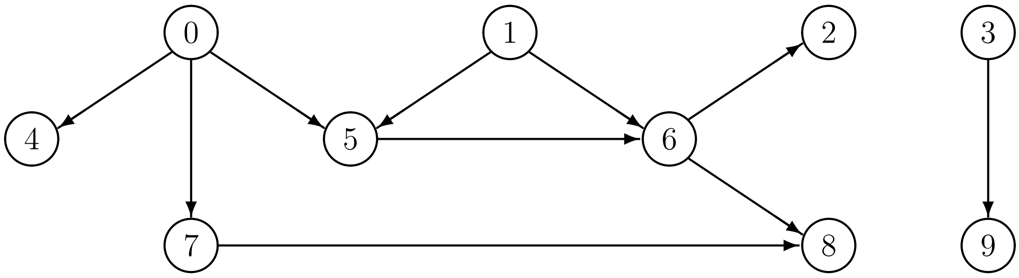
Departamento de Informática, FCT NOVA

21 de Abril de 2017

Responda a **perguntas** diferentes em **folhas** diferentes.

Se precisar de folhas, peça ao docente.

1. [3 valores] Suponha que se executa a **versão recursiva** do algoritmo *dfsTraversal* (que percorre os vértices de um grafo em profundidade) com o grafo G esquematizado na figura.



Assuma que os métodos *nodes* e *outAdjacentNodes* iteram sempre os vértices por ordem crescente. Por exemplo, $G.outAdjacentNodes(0)$ produz os vértices 4, 5 e 7 (por esta ordem).

Indique a ordem pela qual os vértices são percorridos (ou processados). Um vértice v é processado quando é executada a atribuição `processed[`v`] = true`.

2. [6 valores] Considere a seguinte função recursiva, $f_n(i, j)$, onde n, i e j são números inteiros tais que: $n \geq 2, \quad i \geq 1 \quad \text{e} \quad 0 \leq j \leq n$.

$$
f_n(i,j) = \begin{cases} 1, & \text{se } i = 1 \text{ e } 0 \leq j \leq 1; \\ 0, & \text{se } i = 1 \text{ e } 2 \leq j < n; \\ \displaystyle\sum_{k=0}^{\min(i,n-1)} f_n(i,k), & \text{se } i \geq 1 \text{ e } j = n; \\ f_n(i-1, n), & \text{se } i \geq 2 \text{ e } j = 0; \\ f_n(i-1, j-1), & \text{se } i \geq 2 \text{ e } 1 \leq j < n. \end{cases}
$$

Note que o valor do parâmetro n não varia entre chamadas recursivas. Por esse motivo, optou-se por escrever $f_n(i, j)$ em vez de $f(n, i, j)$.

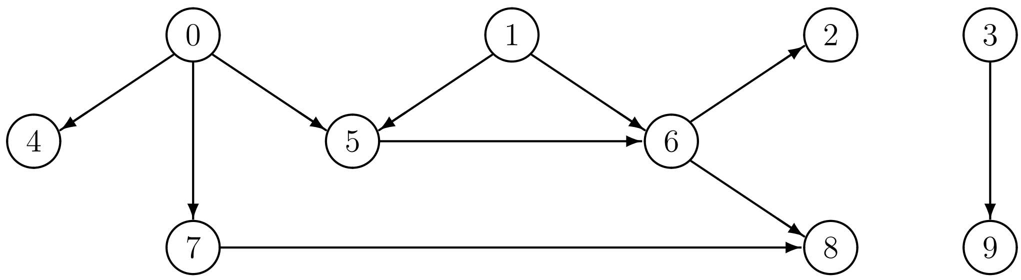
Apresente um algoritmo iterativo, desenhado segundo a técnica da programação dinâmica e implementado em Java, que recebe dois inteiros positivos, c e n, tais que:

$$c > n \geq 2$$

e calcula o valor de $f_n(c, n)$. Estude (justificando) as complexidades temporal e espacial do seu algoritmo.

3. Suponha que quer instalar vários pacotes do sistema de operação Debian, que tem mais de 43 000 pacotes. Sabe que alguns pacotes só podem ser instalados se outros já estiverem instalados e que os pacotes têm de ser instalados sequencialmente (em cada momento, apenas um pacote poderá estar a ser instalado).

   Para facilitar o processo de instalação dos pacotes, decidiu considerar um grafo orientado e acíclico, onde cada vértice representa um pacote. A existência de um arco do vértice v para o vértice w indica que o pacote v terá de estar instalado antes de se iniciar a instalação de w. Agora, perante dois pacotes concretos (p e q), pretende saber se os pode instalar por qualquer ordem (p antes ou depois de q) ou se, pelo contrário, um deles tem de ser instalado antes do outro.



   Para exemplificar, considere o grafo de pacotes esquematizado na figura.

   - **Pacotes 1 e 7:** Os pacotes 1 e 7 podem ser instalados por qualquer ordem.
     Justificação: Por exemplo, 0 4 7 1 5 6 2 8 3 9 e 0 1 7 4 5 6 2 8 3 9 são duas ordens possíveis de instalação dos pacotes. Na primeira, o pacote 7 é instalado antes do pacote 1; na segunda, 1 é instalado antes de 7.
   - **Pacotes 1 e 8:** O pacote 1 tem de ser instalado antes do pacote 8.
     Justificação: O pacote 8 só pode ser instalado se 6 já estiver instalado e a instalação de 6 só pode ser efetuada após a instalação de 1.

   (a) [5 valores] Apresente uma função booleana (em pseudo-código) que recebe:

   - um grafo G orientado e acíclico, com a informação sobre os pacotes a instalar e as suas dependências, e
   - dois pacotes distintos, p e q,

   e retorna *true* se, e só se, p e q puderem ser instalados por qualquer ordem.

   (b) [1 valor] Que estruturas de dados usaria para implementar o grafo? Não escreva código, mas pode ilustrar a sua resposta com o grafo do exemplo.

   (c) [1 valor] Estude (justificando) a complexidade temporal do seu algoritmo, no pior caso, assumindo que o grafo está implementado como indicou na alínea anterior.

4. [4 valores] O Jogo dos Feijões é jogado por dois jogadores que retiram, alternadamente, um montinho (completo) de feijões de uma das extremidades de uma sequência de montinhos de feijões. O jogo acaba quando é retirado o último montinho de feijões e o vencedor é o jogador que conseguiu acumular mais feijões no decorrer do jogo. Se os dois jogadores tiverem retirado o mesmo número de feijões, há empate.

Sequência inicial: 5 8 3 1							
Feijões retirados				Total de feijões		Pontuação de A	Vencedor
A	B	A	B	Total(A)	Total(B)	Total(A) − Total(B)	
5	8	3	1	8	9	-1	B
5	1	8	3	13	4	9	A
1	5	8	3	9	8	1	A
1	3	8	5	9	8	1	A

Considere, para exemplificar, que a sequência inicial tem quatro montes, com **5**, **8**, **3** e **1** feijões, por esta ordem. A tabela acima mostra algumas jogadas possíveis dos jogadores, onde A é o jogador que retira o primeiro monte de feijões e B representa o outro jogador.

O jogador A pode começar por retirar o monte com **5** feijões ou o monte com **1** feijão.

- Se A retirar o monte com **5** feijões, o jogador B, na sua vez, pode optar por retirar o monte com **8** feijões ou o monte com **1** feijão.

  Se B optar pelo monte com **8**, A poderá retirar o monte com **3** feijões, deixando o monte com **1** único feijão para a última jogada de B.

  Se for esta a sequência de jogadas (A retira **5**, B retira **8**, A retira **3**, B retira **1**), **a pontuação de** A, que é a diferença entre o número total de feijões retirados por A e o número total de feijões retirados por B, será $(5+3)-(8+1) = -1$. Como a pontuação de A é negativa, B vence o jogo.
- Se A, na sua primeira jogada, retirar o monte com **1** feijão, então, quer B escolha o monte com **3** ou o monte com **5** feijões, A poderá retirar o monte com **8** feijões na jogada seguinte. Em ambos os casos, como se mostra nas duas últimas linhas da tabela, a pontuação de A é 1 e A vence o jogo.

Dada a sequência inicial de montes de feijões, pretende-se descobrir a maior pontuação que o primeiro jogador (A) pode obter, assumindo que o outro jogador (B) também faz as melhores jogadas possíveis durante o jogo. Para a sequência (**5 8 3 1**), se os dois jogadores jogarem da melhor forma possível, a maior pontuação que o primeiro jogador pode obter é 1.

Apresente **uma função matemática recursiva** que, com base numa sequência de números inteiros positivos, que correspondem à quantidade de feijões nos montinhos,

$$S = (q_1\ q_2\ \cdots\ q_n) \qquad (\text{com } n \geq 2),$$

calcule a maior pontuação que o primeiro jogador pode obter, assumindo que os dois jogadores jogam da melhor forma possível. Indique claramente o que representa cada uma das variáveis que utilizar e explicite a chamada inicial (a chamada que resolve o problema).