



UNIVERSIDADE DE COIMBRA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Estratégias Algorítmicas
Exame Época Normal – 9 de junho de 2025

Nome: _____ Nº de estudante: _____

15 pontos no total, 2 horas, sem consulta.

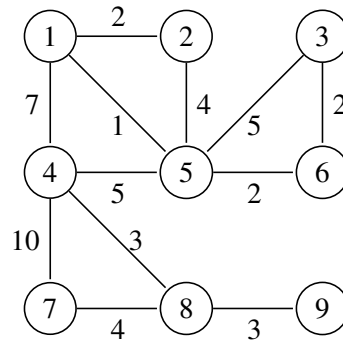
1. A estrutura *union-find* utiliza o método *find(a)* para identificar a raiz da árvore à qual o elemento *a* pertence.

- a) Dado o seguinte pseudo-código que implementa uma variante desse método, e considerando o vetor $P = [2, 0, 4, 2, 4, 2]$, indique qual será o conteúdo do vetor após a execução de *find(1)*. Assuma que *P* está declarado como uma variável global e que seu primeiro elemento tem índice 0 (1.5 pontos).

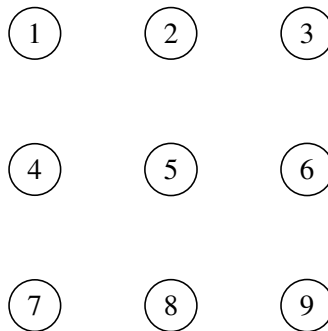
```
Function find(a)  
  if  $P[a] \neq a$  then  
     $P[a] = \textit{find}(P[a])$   
  return  $P[a]$ 
```

- b) O pseudo-código do método *find* na alínea a) é recursivo. Escreva o pseudo-código do mesmo método em versão iterativa. Indique a sua complexidade computacional no pior caso se o *P* tiver *n* elementos (2 pontos).

2. Considere o seguinte grafo.



- a) Desenhe o caminho mais curto do nó 9 a cada nó do grafo usando o algoritmo de Dijkstra, indicando a distância mais curta em cada nó. Assuma que pode ir em ambos os sentidos em cada aresta. Indique os nós visitados pela ordem que o algoritmo os seleciona (2 pontos).



Nós visitados:

- b) Considere o problema de encontrar o caminho entre dois nós num grafo em que a distância máxima associada a um arco ou aresta nesse caminho é a menor possível. Que alteração no algoritmo de Dijkstra deve efetuar para encontrar esse caminho (1.5 pontos)?

3. Considere a seguinte recorrência para $i = 0, \dots, n$ e $c_i \geq 0$:

$$S(i) = \begin{cases} c_i & \text{se } i = 0 \text{ ou } i = 1 \\ c_i + \min\{S(i-1), S(i-2)\} & \text{caso contrário} \end{cases}$$

- a) Tendo em conta que o valor que pretende obter é retornado por $\min\{S(n-2), S(n-1)\}$, apresente o pseudo-código de um algoritmo de programação dinâmica descendente (*top-down*) que explore a recorrência acima para obter esse valor. (1.5 pontos)

- b) Apresente de seguida o pseudo-código de um algoritmo de programação dinâmica ascendente (*bottom-up*) que explore a recorrência acima para obter esse valor e discuta a sua complexidade computacional e espacial. (1.5 pontos)

- c) Discuta possíveis melhorias na sua abordagem apresentada na alínea b), com foco na otimização do uso de memória (1 ponto).

4. Considere uma matriz M de tamanho 3×3 e um inteiro k , com $0 \leq k \leq 4$.

- a) Escreva o pseudo-código de um algoritmo recursivo que imprima todas as configurações distintas possíveis de colocar k símbolos x em posições adjacentes (cima, baixo, esquerda, direita) à posição central de M . Cada configuração deve conter exatamente k símbolos colocados em posições distintas e adjacentes à posição central. Assuma que k e a matriz M estão declaradas como variáveis globais, e que existe um método $print(M)$ que imprime o conteúdo da matriz M . Adicione comentários ao seu código (3 pontos).

- b) Qual é a complexidade temporal da sua abordagem, em termos do número de chamadas recursivas, em função de k ? Justifique a sua resposta (1 ponto).

