

4.0 **1:** Uma procura em profundidade (*depth-first*) é geralmente mais fácil de implementar que uma em largura (*breadth-first*). Explique porquê, e indique as principais vantagens e desvantagens de cada destas duas maneiras de efetuar uma pesquisa exaustiva.

4.0 **2:** Num cubo $n \times n \times n$, em que as coordenadas começam em 0 e acabam em $n - 1$, pretende-se contar o número de maneiras de ir da posição $(0, 0, 0)$ até à posição $(n - 1, n - 1, n - 1)$. Os movimentos válidos são deslocamentos de uma única posição numa das coordenadas, sempre na direção do destino:

$$(\Delta x, \Delta y, \Delta z) = (1, 0, 0) \text{ ou } (0, 1, 0) \text{ ou } (0, 0, 1).$$

Além disso, as coordenadas de todos os pontos do caminho têm de satisfazer as condições

$$0 \leq z \leq y \leq x < n.$$

1.0 a) Que técnica algorítmica é recomendada para resolver este problema?

2.0 b) Escreva código para resolver este problema (para um valor de n fixo, por exemplo 10).

1.0 c) Qual a complexidade computacional da sua solução?

4.0 **3:** Explique como funciona o *merge sort*. Diga também quais são os melhores e piores casos para este algoritmo de ordenação, e mostre que a sua complexidade computacional é $O(n \log n)$. Use o *master theorem*, que afirma que se $T(n) = aT(n/b) + f(n)$ então

- se $f(n) = O(n^{\log_b a - \epsilon})$ para um $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$,
- se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = O(n^{\log_b a} \log n)$, ou
- se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para um $\epsilon > 0$ e se $af(\frac{n}{b}) \leq cf(n)$ para $c < 1$ e n suficientemente grande, então $T(n) = \Theta(f(n))$.

4.0 **4:** Um grafo tem vértices numerados de 1 a n . Existe uma aresta entre os vértices i e j se i for múltiplo de j ou se j for múltiplo de i (considere que um número não é múltiplo de si próprio).

1.0 a) Para $n = 6$, desenhe o grafo.

1.0 b) Represente o grafo da alínea a) usando uma matriz de adjacência.

1.0 c) Represente o grafo da alínea a) usando listas de adjacência.

1.0 d) Para um n genérico, qual das duas representações é melhor?

4.0 **5:** A técnica de programação dinâmica pode ser usada para resolver o problema do caixeiro viajante (*traveling salesman problem*). Explique como, e indique qual a sua complexidade computacional.