

15 de Dezembro de 2017

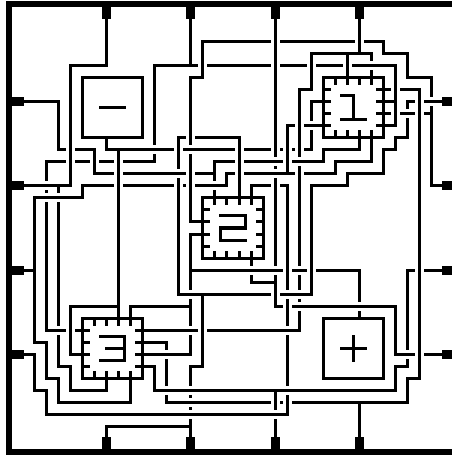
Para ser entregue em papel, manuscrito pela sua própria mão, até ao dia 20 de Dezembro às 16h

**Justifique todas as suas respostas.**

---

4.0 **1:** Seja  $n$  o seu número mecanográfico e seja  $F_n$  o  $n$ -ésimo número de Fibonacci (ver página 4 da aula TP.01). Faça um programa que calcule o resto da divisão de  $F_n$  por  $2n + 1$ . (Apresente o programa e o seu resultado.) Que estratégia algorítmica usou?

5.0 **2:** Considere o labirinto recursivo da figura seguinte.



Neste labirinto os quadrados com os números **1**, **2** e **3** são cópias do quadrado todo, ou seja, dentro de cada um desses quadrados está o quadrado todo (daí o labirinto ser recursivo). Pretende-se ir do quadrado  $+$  para o quadrado  $-$ , **no mesmo nível**, isto é, a ordem por que se sai dos quadrados **1**, **2** e **3** tem de ser a ordem inversa por que se entra nesses mesmos quadrados. Por exemplo, uma ordem possível de entradas e saídas é: entra em 1, entra em 3, entra em 2, sai de 2, entra em 1, sai de 1, sai de 3, sai de 1. (No nível exterior não se pode sair do quadrado.)

Descreva, **por palavras**, um algoritmo capaz de resolver o problema. (Não é preciso fazer o programa!) Pode usar *depth-first search* para resolver este problema?

5.0 **3:** Considere um rectângulo de largura  $w$  e altura  $h$ . Pretende-se contar o número de maneiras de ir do canto inferior esquerdo (coordenadas  $x = 1$  e  $y = 1$ ) para o canto superior direito (coordenadas  $x = w$  e  $y = h$ ) com um custo que não ultrapassa **10**, fazendo movimentos dos seguintes tipos:

- movimento para a direita ( $\Delta x = 1$ ,  $\Delta y = 0$ ), custo **0**;
- movimento para cima ( $\Delta x = 0$ ,  $\Delta y = 1$ ), custo **0**;
- movimento para a direita e para cima ( $\Delta x = 1$ ,  $\Delta y = 1$ ), custo **0**;
- movimento para a esquerda ( $\Delta x = -1$ ,  $\Delta y = 0$ ), custo **1**;
- movimento para baixo ( $\Delta x = 0$ ,  $\Delta y = -1$ ), custo **1**; e
- salto de cavalo ( $\Delta x = 2$ ,  $\Delta y = 1$ ), custo **2**.

Escreva código **C** capaz de resolver o problema (use o tipo de dados **long** para armazenar números de movimentos.) Que estratégia algorítmica usou, e qual a sua complexidade computacional?

(Continua na próxima página.)

**3.0** **4:** Um grafo tem **5** vértices. Os vértices desse grafo estão numerados de **1** a **5**; entre o vértice número ***i*** e o vértice número ***j*** só existe uma aresta se apenas um desses dois números for um número primo.

- Desenhe o gráfico.
- Represente o grafo pela sua matriz de adjacência.
- Represente o grafo por listas de adjacência.
- Quantas maneiras diferentes existem para se ir do vértice **1** para o vértice **2** sem nunca passar pela mesma aresta duas vezes?

**3.0** **5:** Na resolução de um problema usando a estratégia *divide-and-conquer* chegou-se à conclusão que o tempo de execução ***T(n)*** obedecia à recursão

$$T(n) = 3T\left(\frac{n}{2}\right) + n \log n, \quad n > 1,$$

com  $T(0) = T(1) = 1$ . A que ritmo cresce ***T(n)***?