



Exercícios para entregar

1. Uma maneira de encontrar maior divisor comum MDC de dois inteiros x e y é dado por

$$\text{mdc}(x, y) = \begin{cases} x, & \text{se } x = y \\ 2 * \text{mdc}(\frac{x}{2}, \frac{y}{2}), & \text{se } x \text{ é par e } y \text{ é par} \\ \text{mdc}(\frac{x}{2}, y), & \text{se } x \text{ é par e } y \text{ é ímpar} \\ \text{mdc}(x, \frac{y}{2}), & \text{se } x \text{ é ímpar e } y \text{ é par} \\ \text{mdc}(\frac{x-y}{2}, y), & \text{se } x \text{ é ímpar e } y \text{ é ímpar e } x > y \\ \text{mdc}(x, \frac{y-x}{2}), & \text{se } x \text{ é ímpar e } y \text{ é ímpar e } x < y \end{cases}$$

(a) Escreva uma função recursiva para computar o $\text{mdc}(x, y)$.

(b) Verifique suas funções para as seguintes entradas:

$$\text{mdc}(270, 192) = 6$$

$$\text{mdc}(35, 10) = 5$$

$$\text{mdc}(10, 15) = 5$$

$$\text{mdc}(31, 2) = 1$$

2. Um *bitonic array* é um vetor de números que é organizado em duas partes. A primeira parte que consiste de elementos na ordem crescente e a segunda parte consiste de elementos em ordem decrescente. Mais precisamente, uma vetor $A[1..n]$ é *bitonic* se e somente se existe um índice i , $1 \leq i \leq n$, tal que $A[1..i]$ é estritamente crescente, e $A[i..n]$ é estritamente decrescente. Por exemplo, o vetor $\{2, 5, 8, 7, 3\}$ é *bitonic*, com $i = 3$. Faça uma função, utilizando a técnica de divisão e conquista que encontre o máximo elemento de um *bitonic array* (assuma que todos os elementos são distintos).
3. O cálculo da raiz quadrada de um número real x pode ser feito através do seguinte algoritmo:

$$\text{RaizQ}(x, x_0, \varepsilon) = \begin{cases} x_0 & \text{se } |x_0^2 - x| \leq \varepsilon \\ \text{RaizQ}(x, \frac{x_0^2 + x}{2x_0}, \varepsilon) & \text{caso contrário} \end{cases}$$

em que x_0 é uma aproximação inicial do valor \sqrt{x} e ε é um erro admissível. Implemente o algoritmo em linguagem C e mostre o diagrama de execução para a chamada $\text{RaizQ}(13, 3.2, 0.001)$.

4. Dada uma sequência de números inteiros com n elementos, determinar quantas subsequências de números iguais consecutivos compõem essa sequência. Exemplos:
- a) A sequência 5 2 2 3 4 4 4 4 1 1, é composta por 5 subsequências:
(5), (2, 2), (3), (4, 4, 4, 4), (1, 1);
- b) A sequência: 3 3 - 1 - 1 - 1 12 12 12 3 3 é composta por 4 subsequências:
(3, 3), (-1, -1, -1), (12, 12, 12), (3, 3);



Escreva uma função na linguagem C que determina a quantidade de subsequências em sequência dada. Qual é complexidade da sua função ?

5. Dada uma sequência de números inteiros com n elementos, faça uma função para determinar o comprimento de um segmento crescente de comprimento máximo.

Exemplos: Na sequência 5 10 3 2 4 7 9 8 5 o comprimento do segmento crescente máximo é 4 (2, 4, 7, 9).

Na sequência 10 8 7 5 2 o comprimento de um segmento crescente máximo é 1.

Qual é complexidade da sua função ?

6. Dada a sequência $a[1..n]$ de números inteiros, escreva um programa na linguagem C que verifica se existem dois **segmentos** consecutivos iguais na sequência, isto é, se existem i e m tais que:

$$a[i] \ a[i+1] \ \cdots \ a[i+m-1] = a[i+m] \ a[i+m+1] \ \cdots \ a[i+2m-1]$$

Imprima, caso existam, os valores de i e m .

Exemplo: Na sequência 7 9 5 4 5 4 8 6 existem $i = 3$ e $m = 2$.

Qual é complexidade da sua função ?

7. Dada uma sequência de números inteiros com n elementos, escreva uma função na linguagem C para determinar um **segmento** de soma máxima.

Exemplo: Na sequência 5 2 -2 -7 3 14 10 -3 9 -6 4 1, a soma do segmento é 33. Qual é complexidade da sua função ?

8. Dados dois números naturais m e n e duas **seqüências ordenadas** sem números repetidos com m e n números inteiros, obter uma única sequência ordenada contendo todos os elementos das seqüências originais. Qual é complexidade da sua função ?

9. Escreva uma função recursiva que apresente números binários de tamanho n .

- Exemplo: $n = 3$

Saída :

000
001
010
011
100
101
110
111



10. Considere uma partida de futebol entre duas equipes $A \times B$, cujo placar final é $m \times n$, em que m e n são número de gols marcados por A e B, respectivamente. Escreva um algoritmo recursivo que imprima todas as possíveis sucessões de gols marcados. Por exemplo, para um placar final de 3×1 , as possíveis sucessões de gols são “AAAB”, “AABA”, “ABAA” e “BAAA”.
11. Escreva um algoritmo utilizando a estratégia Backtracking que ordena um vetor de N elementos inteiros. (Um tanto ridículo, mas pode ser um bom exercício). O seu algoritmo funciona para números repetidos ?
12. (QUADRADO COM PALITOS DE FÓSFOROS) Dado um vetor de inteiros *palitos_de_fosforos*, onde o *palitos_de_fosforos[i]* é o tamanho do i-esimo palito de fósforo. Deseja-se usar todos os palitos de fósforos para fazer um quadrado. Não é permitido quebrar nenhum palito. Escreva uma função que devolva true se você pode construir um quadrado e falso caso contrário.
Restrições:
 - $1 \leq \text{len}(\text{palitos_de_fosforos}) \leq 15$
 - $1 \leq \text{palitos_de_fosforos}[i] \leq 10^8$
13. (PERCURSO DO CAVALO) Dado um tabuleiro com $n \times n$ posições, o cavalo movimenta-se segundo as regras do xadrez. A partir de uma posição inicial (x0, y0), o problema consiste em fazer o cavalo “visitar” todas as casas do tabuleiro, sem repetições.

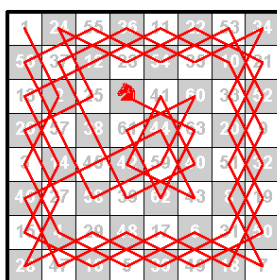


Figura 1: Percurso do cavalo no tabuleiro de xadrez

Escreva um programa que selecione uma determinada posição do tabuleiro e verifique se é possível realizar o percurso do cavalo no tabuleiro de xadrez.

14. (URI Online Judge | 1055) Soma Permutada Elegante
São disponibilizados a você n inteiros $A_1 A_2 A_3 \dots A_n$. Encontre uma permutação destes n inteiros de forma que a soma da diferença absoluta entre os elementos adjacentes seja maximizada.



Supondo $n = 4$ e que os inteiros fornecidos sejam 4 2 1 5. A permutação 2 5 1 4 permite a máxima soma. Para esta permutação, teríamos soma = $\text{abs}(2-5) + \text{abs}(5-1) + \text{abs}(1-4) = 3+4+3 = 10$. De todas as 24 permutações possíveis, você não vai obter qualquer soma cujo valor exceda 10. Nós iremos chamar este valor 10, de soma permutada elegante.

Escreva uma função, usando Backtracking, que dado n inteiros encontre o valor da permutada elegante.

15. SUBSET SUM - O PROBLEMA DO CHEQUES:

Suponha que p_1, p_2, \dots, p_n são os valores dos cheques que você emitiu durante o mês. No fim do mês, o banco debita uma quantia C na sua conta. Você quer saber se algum conjunto de cheques corresponde ao valor debitado. Considere que não temos cheques com valores repetidos. Generalização do problema: Dado um conjunto de números naturais $P = \{p_1, p_2, \dots, p_n\}$ e C , decidir se existe um subconjunto X de $\{1, 2, \dots, n\}$ tal que o somatório dos elementos P relacionados em X é igual a C .

Exemplo1: $P = \{30, 40, 10, 15, 11, 60, 54\}$ e $C = 55$, teríamos os subconjuntos

- $X = \{2, 4\}$ $p_2 = 40$ $p_4 = 15$
- $X = \{1, 4, 5\}$, $p_1 = 30$, $p_4 = 15$, $p_5 = 10$