

1. Deﬁna algoritmo.

Seqüência de ações (passos) finitas,

bem definidas, que resolve,

em tempo finito,

um problema de ordem computacional.

1. Das formas de representar algoritmos explique, justiﬁcando, qual a mais adequada para o desenvolvimento de programas.

Das formas de representação:

* Textual
* fluxograma
* pseudocodigo

A que mais fácil você poder representar aquele programa, transformando de ideias, para o papel (ou programa).

1. Explique o que é **eﬁciência** de um algoritmo e porque seu estudo é importante para o desenvolvimento de programas de computador.

A eficiência leva em conta fatores como tempo de processamento e memória usada para definir um bom algoritmo

1. Considere o conjunto de números naturais. Uma das formas de determinar se um número natural *n* é primo é contar quantos divisores tem *n* no conjunto. Escreva um algoritmo que receba um número natural *n* e retorne se *n* é primo ou não. Seu algoritmo deve contar quantos divisores tem *n* e usar essa informação para determinar se *n* é primo.

1. Considere um *array A* de *n* números naturais *Ai*. Escreva um algoritmo que determine a maior diferença absoluta um par de elementos consecutivos da lista.

O algoritmo deve retornar o valor da diferença.

1. Considere um *array A* de *n* números naturais *Ai* e *s* um valor natural. Escreva um algoritmo que verﬁque se existe um par de elementos em *A* cuja soma seja *s*.

O algoritmo deve retornar um valor lógico (verdadeiro ou falso) que determinar se tal soma existe.

1. Considere um *array A* de *n* números naturais *Ai* e *s* um valor natural. Escreva um algoritmo que verﬁque quantos pares de elementos em *A* cuja soma seja *s* existem.

O algoritmo deve retornar um valor inteiro, a quantidade de pares cuja soma seja *s*

1. Escreva um algoritmo que receba um *array A* de *n* números naturais e determine se o *array* está ordenado em ordem não decrescente, ou seja, do menor para o maior.