

Algoritmos

A modern beamer theme

Prof.: Maiquel de Brito

13 de julho de 2018

BLU3101 - Introdução à Informática para Automação

Departamento de Engenharias

UFSC Blumenau

Table of contents

1. Introdução
2. Algoritmos

Introdução

Ábaco; Charles Babbage; Ada Lovelace; George Boole; Hollerith; MARK I; ENIAC; John von Neumann; Alan Turing; Mainframes; Inteligência Artificial; IBM; Oracle; Bill Gates; Microsoft; Steve Jobs; Apple; Computador Pessoal; Internet; Google; Computação em Nuvem; Redes Sociais; Computação Embarcada; Dispositivos Móveis; Internet das Coisas (IoT); Deep Learning; Veículos Autônomos; ...

Algoritmos

- As soluções computacionais para os diferentes problemas são construídas através de **algoritmos**.
- *Sequência finita de ações para transformar um determinada entrada em uma determinada saída;*
- Um **algoritmo** é *uma sequência de passos que visam atingir um objetivo bem definido.*
- Ex.: receita de bolo

Exercício

Elaborar um algoritmo que encontre a idade da pessoa mais velha em um grupo de n pessoas.

Exercício - Solução 1

- 1 seja P um conjunto de n pessoas: $P = \{p_1, \dots, p_n\}$
- 2 pergunta a idade de p_1
- 3 anota a idade de p_1
- 4 pergunta a idade de p_2
- 5 se idade de p_2 maior que a idade anotada
- 6 apaga a idade anotada
- 7 anota a idade de p_2
- 8 pergunta a idade de p_3
- 9 se idade de p_3 maior que a idade anotada
- 10 apaga a idade anotada
- 11 anota a idade de p_3
- 12 \vdots
- 13 pergunta a idade de p_n
- 14 se idade de p_n maior que a idade anotada
- 15 apaga a idade anotada
- 16 anota a idade de p_n

Exercício

- 1 seja P um conjunto de n pessoas: $P = \{p_1, \dots, p_n\}$
- 2 pergunta a idade de p_1
- 3 anota a idade de p_1
- 4 para k de 2 até n
- 5 pergunta a idade de p_k
- 6 se idade de p_k maior que a idade anotada
- 7 apaga a idade anotada
- 8 anota a idade de p_k

Exercício

- 1 seja P um conjunto de n pessoas: $P = \{p_1, \dots, p_n\}$
- 2 anota a idade do mais velho entre p_1 e o mais velho de $\{p_2, \dots, p_n\}$
- 3

$$\text{mais_velho}(\{p_1, \dots, p_n\}) = \begin{cases} p_1 & \text{se } p_1 > p_n \text{ e } n = 2 \\ p_n & \text{se } p_1 \leq p_n \text{ e } n = 2 \\ \text{mais_velho}(\{p_1\} \cup \text{mais_velho}(\{p_2, \dots, p_n\})) & \end{cases}$$

Exercício

São dados N azulejos de dimensões 10cm x 10cm. Com eles, deve-se montar um conjunto de quadrados de modo a utilizar todos os azulejos dados. Inicialmente você deve montar o maior quadrado possível com os azulejos dados; então, com os azulejos que sobraram, deve-se montar o maior quadrado possível, e assim sucessivamente. Por exemplo, se forem dados 31 azulejos, o conjunto montado terá quatro quadrados, conforme ilustra a figura abaixo.



Conjunto com quatro quadrados,
montado a partir de 31 azulejos

Elaborar um algoritmo para calcular quantos quadrados de k azulejos podem ser montados quando é dado um total de N azulejos. No exemplo acima, para 31 azulejos, o programa deve imprimir o seguinte resultado: