

# Complexidade de Algoritmos

Paulino Ng

2020-05-29

# Plano da aula

Esta aula apresenta alguns algoritmos de ordenação e uma análise de sua complexidade.

1. Problema de ordenação.
2. Ordenação por seleção
3. Ordenação por inserção
4. *Mergesort*, ordenação por intercalação
5. *Quicksort*

# Problema da Ordenação

*Entrada:*

*Saída:*

# Ordenação por seleção

Procedimento `ordena_selecao(A,n)`

Entradas:

A: um vetor.

n: número de elementos de A a serem ordenados.

Saída: Os elementos de A ordenados em ordem não decrescente

1. Para  $i = 0$  até  $n-2$ :

A. Faça `menor` ser o índice do menor elemento do subvetor

B. Troque `A[i]` com `A[menor]`

Código Python para implementar este algoritmo

# Ordenação por inserção

Procedimento ordena\_insercao(A,n)

Entradas e Saída como no ordena\_selecao()

1. Para  $i=1$  até  $n-1$ :
  - A. Faça chave =  $A[i]$  e  $j = i - 1$
  - B. Enquanto  $j > 0$  e  $A[j] > \text{chave}$  faça:
    - i.  $A[j+1] = A[j]$
    - ii.  $j = j - 1$
  - C.  $A[j+1] = \text{chave}$

# Mergesort

# Quicksort