

# MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

## Ordenação – algoritmos quadráticos

## Ordenação

Considere um vetor com  $n \geq 0$  elementos (vamos considerar que são números inteiros, mas poderia ser qualquer tipo de dado). O problema que desejamos resolver é de ordenar este vetor em ordem crescente, ou seja, após a execução do algoritmo,

$$v[i] \leq v[i + 1], \text{ para } i = 0, \dots, n - 1.$$

Exemplo: Dado o vetor,

44	75	12	27	95	-15	3	73	19	8	12	22
----	----	----	----	----	-----	---	----	----	---	----	----

a saída deverá ser

-15	3	8	12	12	19	22	27	44	73	75	95
-----	---	---	----	----	----	----	----	----	----	----	----

## Bubblesort

**Ideia:** Percorre o vetor enquanto encontrar **inversões**. Se não houver inversões, o vetor estará ordenado. Caso contrário, resolve a inversão e itera novamente.

Uma **inversão** é um par de índices  $0 \leq i < j \leq n - 1$  do vetor tais que  $v[i] > v[j]$ .

Observe que, se houver pelo menos uma inversão no vetor, então haverá uma inversão entre dois elementos vizinhos.

## Bubblesort - algumas observações

- ▶ O algoritmo está correto? Como provar?
- ▶ Qual o **invariante**? O que é verdade em toda iteração do algoritmo?
- ▶ Versão recursiva?
- ▶ Quantas comparações e trocas entre elementos do vetor são feitas?
- ▶ Instâncias ruins? *Turtles and rabbits*?
- ▶ Animação do algoritmo

## Ordenação por seleção

**Ideia:** Em cada iteração seleciona o maior (menor) elemento do intervalo e coloca na última (primeira) posição.

**Invariante:** No fim da iteração  $i$ , temos:



Além disso, para todo  $x, y$  com  $x \in P_1$  e  $y \in P_2$ ,  $x \leq y$ .

Quantas comparações? Quantas trocas? Versão recursiva?

Animação do algoritmo