ICMC - Instituto de Ciências Matemáticas e de Computação

BCC - Bacharelado em Ciências da Computação

Disciplina: Laboratório de Introdução a Ciência da Computação

**Docente**: Leonardo Pereira

Aluno: Bernardo Marques Costa (Número USP 11795551)

## ANÁLISE DE ALGORITMO E COMPLEXIDADE

## INTRODUÇÃO

O intuito deste trabalho é apresentar o tempo de execução para dois algoritmos simples: **bubblesort** e **inversor de arrays.** Será apresentado os algoritmos em si e os gráficos construídos com os seguintes dados: número de inputs fornecidos (tamanho do vetor que sofrerá ação do algoritmo) e o tempo de execução (tempo médio que o algoritmo levou para processar o input fornecido, foram utilizadas 1000 execuções para tornar um valor médio aceitável).

## **ALGORITMOS E CONTAGEM DE OPERAÇÕES**

#### Bubblesort

#### Inverter array

### **OBTENDO OS DADOS**

Para executar a análise dos dados, foi utilizado um programa que requisitava a seguinte linha de comando para rodar o binário e salvar os dados no arquivo

```
./<programa> <Nome da função desejada> <Número de repetições> <Número de dados>
C ∨
```

Caso a linha de comando executada fosse diferente da desejada pelo programa, seria apresentada a seguinte saída de erro:

```
Usage: ./a.out < function > < N executions > < Number of data >
- function available: bubblesort invert
- N executions: integer
- Number of data: 10, 100, 1000, 5000, 10000, 100000, 1000000
```

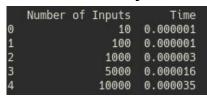
A partir da execução do programa é construído um arquivo **bubblesort.dat** ou **invert.dat**, um arquivo semelhante a um CSV com dados correlacionais: números de inputs e tempo de execução

Obtemos os dois data frame sobre os algoritmos **bubblesort** e **inversor de arrays**, a partir da utilização da biblioteca **pandas** de python:

#### **Bubblesort**

	Number	of	Inputs	Time
0			10	0.000001
1			100	0.000041
2			1000	0.003982
3			5000	0.088282
4			10000	0.375590

#### Inversor de arrays



# **GRÁFICO DO TEMPO DE EXECUÇÃO**

A partir da utilização da biblioteca **matplotlib.pyplot** de python, podemos construir os seguintes dois gráficos, entre número de inputs e tempo (em segundos):

