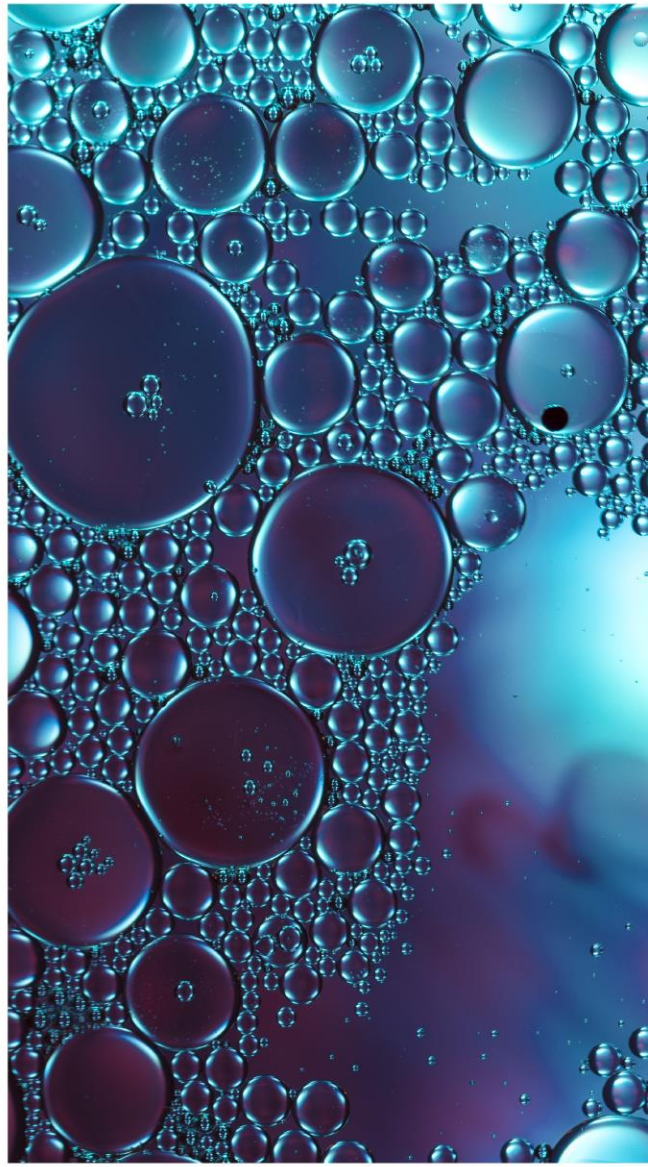
UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA

BUBBLE SORT

ALGORITMOS E ESTRUTURAS DE DADOS II

Profº Dr. Sergio Cleger Tamayo

EQUIPE
Kleyson de Melo Lopes
Lázaro Júnior Machado Pontes



BUBBLESORT

O bubble sort ou método bolha é uma ordenação por flutuação.

Critério de ordenação: **crescente** e **decrescente.**

A filosofia básica deste método consiste em:

1. "varrer" o vetor inteiro comparando elementos adjacentes (dois a dois);
   - v[i] > v[i+1].
2. Caso estejam fora de ordem, os mesmos trocam de posição entre si.

BUBBLESORT

Origem da denominação?

- Os elementos menores vão aos poucos subindo para o início as tabela, essa movimentação lembra a forma como as bolhas em um tanque procuram seu nível.

Onde usar e Como?

- Tabelas muito pequenas,
- Classificação de arquivos pequenos,
- Conjuntos de dados que já se encontram semi-classificados,
- Demonstrações didáticas.

BUBBLESORT

Vantagens

- Simplicidade no algoritmo
- Estável

Desvantagens

- Lento

Complexidade

- Melhor caso: O(N)
- Médio caso: $O(N^2)$
- Pior caso: $O(N^2)$

BUBBLESORT

ITERAÇÕES E TROCAS

X = (25 , 57 , 48 , 37 , 12 , 92 , 86 , 33)

Passo 1

0	25	57	48	37	12	92	86	33
1	25	48	57	37	12	92	86	33
2	25	48	37	57	12	92	86	33
3	25	48	37	12	57	92	86	33
4	25	48	37	12	57	92	86	33
5	25	48	37	12	57	86	92	33
6	25	48	37	12	57	86	33	92

Passo 2

0	25	48	37	12	57	86	33	92
1	25	37	48	12	57	86	33	92
2	25	37	12	48	57	86	33	92
3	25	48	12	48	57	86	33	92
4	25	48	37	12	57	86	33	92
5	25	48	37	12	57	33	86	93
6	25	37	12	48	57	33	86	92

BUBBLESORT ITERAÇÕES E TROCAS

X = (25 , 57 , 48 , 37 , 12 , 92 , 86 , 33)

Passo 3

0	25	37	12	48	57	33	86	92
1	25	12	37	48	57	33	86	92
2	25	12	37	48	57	33	86	92
3	25	12	37	48	57	33	86	92
4	25	12	37	48	33	57	86	92
5	25	12	37	48	33	57	86	92
6	25	12	37	48	33	57	86	92

Passo 4

0	12	25	37	48	33	57	86	92
1	12	25	37	48	33	57	86	92
2	12	25	37	48	33	57	86	92
3	12	25	37	33	48	57	86	92
4	12	25	37	33	48	57	86	92
5	12	25	37	33	48	57	86	92
6	12	25	37	33	48	57	86	92

BUBBLESORT ITERAÇÕES E TROCAS

X = (25 , 57 , 48 , 37 , 12 , 92 , 86 , 33)

Passo 5

0	12	25	37	33	48	57	86	92
1	12	25	37	33	48	57	86	92
2	12	25	33	37	48	57	86	92
3	12	25	33	37	48	57	86	92
4	12	25	33	37	48	57	86	92
5	12	25	33	37	48	57	86	92
6	12	25	33	37	48	57	86	92

passo 0: (vetor original)
25 57 48 37 12 92 86 33
passo 1: 25 48 37 12 57 86 33 92
passo 2: 25 37 12 48 57 33 86 92
passo 3: 25 12 37 48 33 57 86 92
passo 4: 12 25 37 33 48 57 86 92
passo 5: 12 25 33 37 48 57 86 92
passo 6: 12 25 33 37 48 57 86 92
passo 7: 12 25 33 37 48 57 86 92

BUBBLESORT

CÓDIGO IMPLEMENTADO EM C++

Crescente

```
void bubbleSort(int tam, int * v){
 for(int i = 1 ; i < tam ; i++){
   for(int j = 0 ; j < tam-i ;
j++){
     if ( v[j]< v[j-1]){
       int aux = v[j];
       v[j] = v[j-1];
       v[j-1] = aux;
     }
}}
```

Decrescente

```
void bubbleSort(int tam, int * v){
 for( int i = 1 ; i < tam ; i++){
   for( int j = 0 ; j < tam-i ;
j++){
     if ( v[j]> v[j-1]){
       int aux = v[j];
       v[j] = v[j-1];
       v[j-1] = aux;
     }
}}
```

BUBBLESORT

CÓDIGO IMPLEMENTADO EM C++

Crescente Otimizado

```
void bubbleSortOtC(int n, int * v){
 for(int i = 1 ; i < n ; i++){
    int trocado = 0;
   for(int j = 0 ; j < n-i ; j++){
     if ( v[j] > v[j+1]){
       int aux = v[j];
       v[j] = v[j+1];
       v[j+1] = aux;
        trocado = 1;
     }
  }
      if (trocado == 0)
      break;
}}
```

Decrescente Otimizado

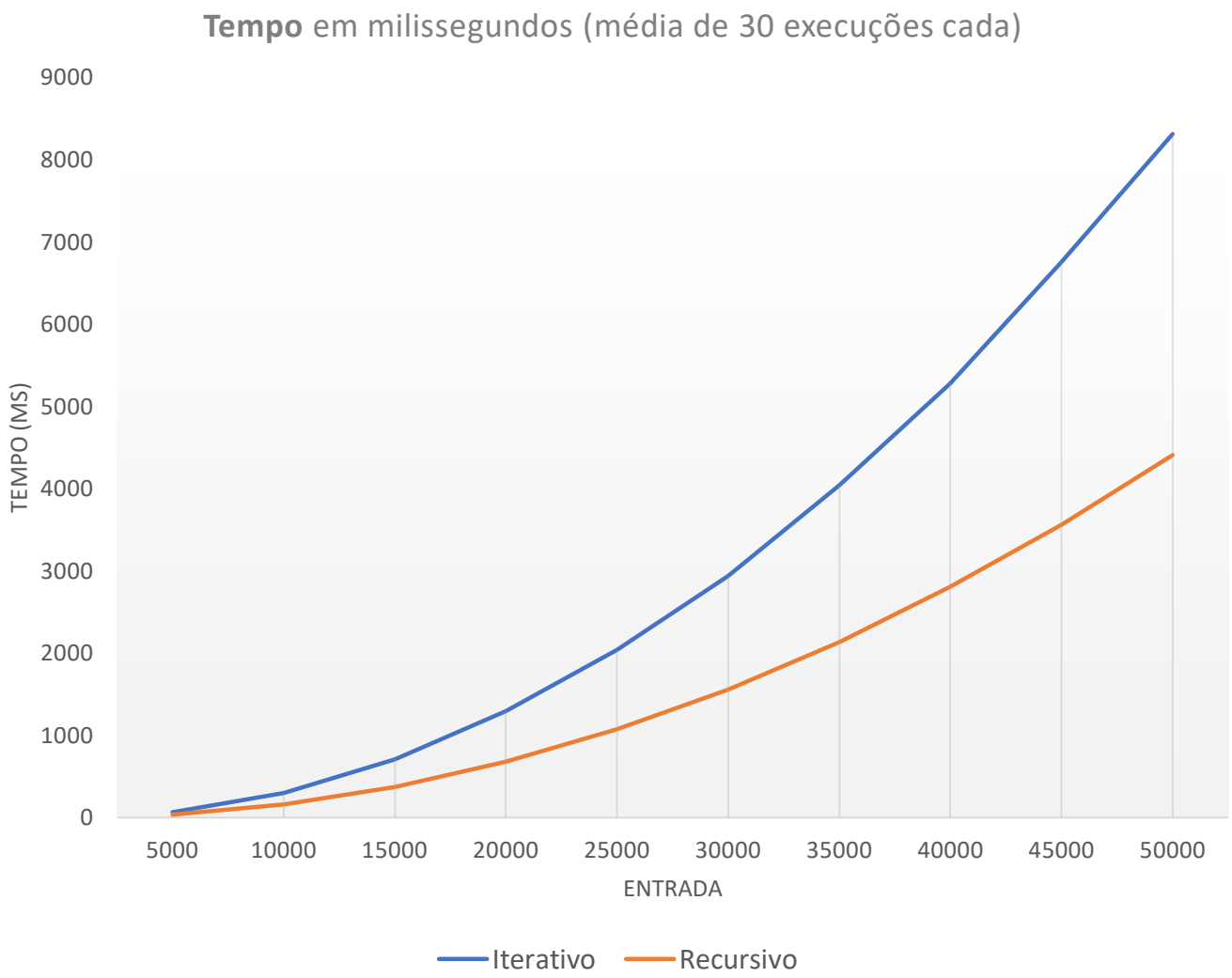
```
void bubbleSortOtD(int n, int * vetor){
 for(int i = 1 ; i < n ; i++){
    int trocado = 0;
   for(int j = 1 ; j < n-i ; j++){
     if ( v[j] < v[j+1]){
       int aux = v[j];
       v[j] = v[j+1];
       v[j+1] = aux;
        trocado = 1;
     }
  }
      if (trocado == 0)
      break;
}}
```

Crescente Recursivo

```
void crescenteRec(int n, int * v){
 if(n == 1)
    return;
   for(int i = 1 ; i < n-1 ; i++){
     if ( v[i] > v[i+1]){
       int aux = v[i];
       v[i] = v[i+1];
       v[i+1] = aux;
     }
   }
     crescenteRec(n-1,v);
}
```

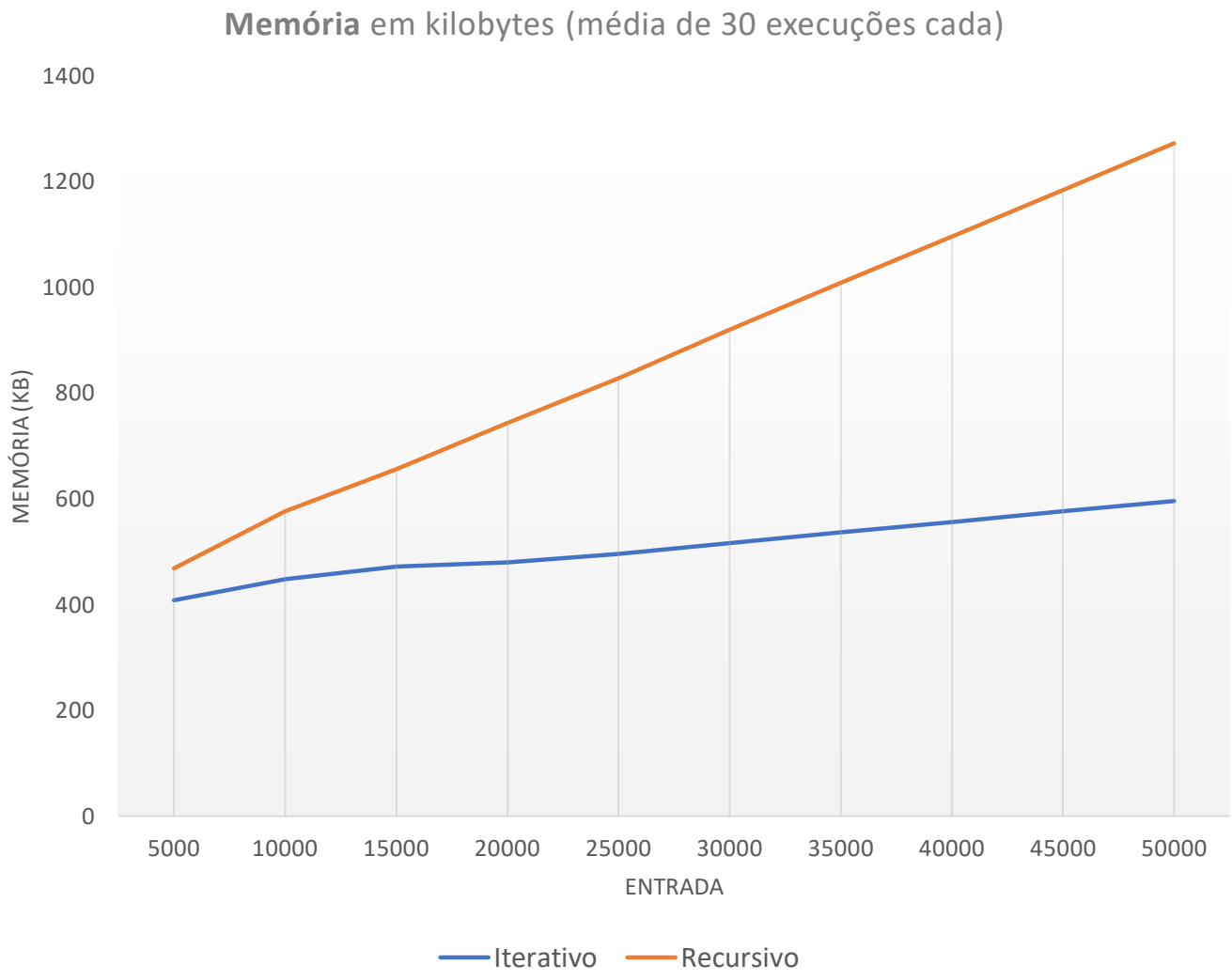
BUBBLESORT

TEMPO DE PROCESSAMENTO



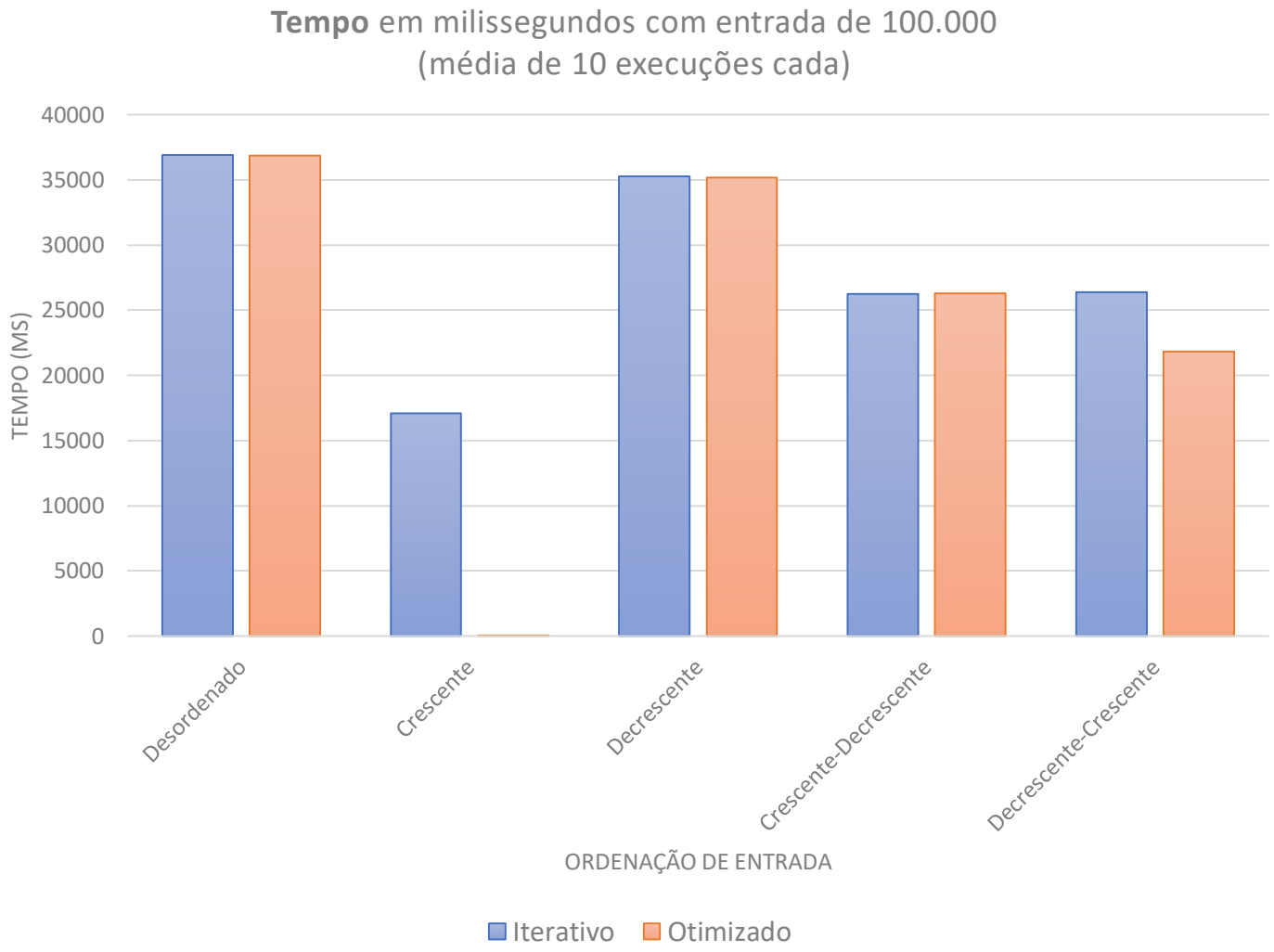
BUBBLESORT

USO DE MEMÓRIA



BUBBLESORT

TEMPO DE PROCESSAMENTO




OBRIGADO