

Engenharia de Computação

Estrutura de Dados 2

Aula 2 – BubbleSort

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

BubbleSort

- Também conhecido como ordenação por bolha
 - Um dos algoritmos mais conhecidos que existem
 - Remete a bolha flutuando em um tanque de água em direção ao topo até atingirem seu próprio nível

BubbleSort

- **Funcionamento:**

- Compara pares de elementos adjacentes e os troca de lugar se estiverem na ordem errada
 - Trabalha de forma a movimentar, uma posição por vez, o maior valor existente na porção não ordenada de um vetor para sua respectiva posição no vetor ordenado
- Esse processo se repete até que mais nenhuma troca seja necessária.
 - Elementos já ordenados

BubbleSort - Pseudocódigo

- BubbleSort (V, n)
 - trocou \leftarrow verdade
 - Enquanto trocou for verdade faça:
 - Trocou \leftarrow Falso
 - Para todo índice i de 0 a $n-2$
 - Se $V[i]$ é maior que $v[i+1]$ então:
 - Troque $V[i]$ com $V[i+1]$
 - trocou \leftarrow verdade

BubbleSort

- Passo a passo:
 - 1ª iteração do-while: encontra o maior valor e o movimenta até a última posição

Sem Ordenar						
	23	4	67	-8	21	
1º Iteração do-while						
i=0	23	4	67	-8	21	$V[i] > V[i+1]$: Trocar
i=1	4	23	67	-8	21	$V[i] < V[i+1]$: Manter
i=2	4	23	67	-8	21	$V[i] > V[i+1]$: Trocar
i=3	4	23	-8	67	21	$V[i] > V[i+1]$: Trocar
Final	4	23	-8	21	67	

BubbleSort

- Passo a passo
 - 2º iteração do-while: encontra o segundo maior valor e o movimenta até a penúltima posição

2º iteração do-while

i=0	4	23	-8	21	67	$V[i] < V[i+1]$: Manter
i=1	4	23	-8	21	67	$V[i] > V[i+1]$: Trocar
i=2	4	-8	23	21	67	$V[i] > V[i+1]$: Trocar
Final	4	-8	21	23	67	

BubbleSort

- Passo a passo
 - Processo continua até todo o array estar ordenado

3º Iteração do-while

i=0

4	-8	21	23	67
---	----	----	----	----

 $V[i] > V[i+1]$: Trocar

i=1

-8	4	21	23	67
----	---	----	----	----

 $V[i] < V[i+1]$: Manter

Final

-8	4	21	23	67
----	---	----	----	----

4º Iteração do-while

i=0

-8	4	21	23	67
----	---	----	----	----

 $V[i] < V[i+1]$: Manter

Não houve mudanças: ordenação concluída

Ordenado

-8	4	21	23	67
----	---	----	----	----

BubbleSort

- Vantagens
 - Simples e de fácil entendimento e implementação
 - Está entre os métodos de ordenação mais difundidos existentes
- Desvantagens
 - Não é um algoritmo eficiente
 - Sua eficiência diminui drasticamente a medida que o número de elementos no array aumenta
 - É estudado apenas para fins de desenvolvimento de raciocínio

BubbleSort - Complexidade

- Considerando um array com N elementos, o tempo de execução é:
 - $O(n)$, melhor caso: os elementos já estão ordenados.
 - $O(n^2)$, pior caso: os elementos estão ordenados na ordem inversa.
 - $O(n^2)$, caso médio.

Exercício

- **1)** Simule o algoritmo utilizando cartas de um baralho ou uno.
- **2)** Implemente o **BubbleSort** em C considerando a seguinte assinatura de função

```
/**  
 * \brief Ordena o vetor usando BubbleSort  
 *  
 * \param v vetor a ser ordenado  
 * \param n tamanho do vetor  
 *  
 * Ordena o vetor usando o método BubbleSort  
 * Esse algoritmo tem um comportamento assintótico  $O(N^2)$   
 */  
void bubbleSort(int *v, int n);
```