

# Pesquisa e Ordenação

## Quadro Resumo

MÉTODO	Complexidade Temporal		Complexidade Espacial	OBSERVAÇÃO
	MELHOR CASO	Pior Caso		
BOLHA (ESTÁVEL)	NÃO TEM MELHOR NEM PIOR CASO. SEMPRE GASTA TEMPO DA ORDEM $\theta(n^2)$		$O(1)$ in situ	Após $n-1$ iterações a LISTA ESTARÁ EM ORDEM. <u>NÃO POSSUI MELHOR NEM PIOR CASO.</u>
BOLHA COM FLAG. (ESTÁVEL)	Em LISTA JÁ ORDENADA $\theta(n)$	menor elemento na última posição. $\theta(n^2)$	$O(1)$	
INSERÇÃO (ESTÁVEL)	LISTA JÁ ORDENADA $\theta(n)$	INVERSAMENTE ORDENADA $\theta(n^2)$	$O(1)$	
SELEÇÃO ( <u>NÃO</u> ESTÁVEL)	NÃO TEM MELHOR NEM PIOR CASO. SEMPRE GASTA TEMPO DA ORDEM $\theta(n^2)$		$O(1)$	Após $n-1$ iterações a LISTA ESTARÁ EM ORDEM
SHELL SORT ( <u>NÃO</u> ESTÁVEL)	LISTA JÁ ORDENADA $\theta(n \log n)$	NÃO SE SABE, MAS ESTÁ ENTRE: $\theta(n \log n)$ e $\theta(n \log^2 n)$	$O(1)$	
MERGESORT (ESTÁVEL)	NÃO TEM MELHOR NEM PIOR CASO $\theta(n \log n)$		$\theta(n)$ , pois utiliza um vetor auxiliar	
MERGE (ESTÁVEL)	O PROCEDIMENTO MERGE REQUER TEMPO DA ORDEM DE $\theta(n)$			
QUICKSORT ( <u>NÃO</u> ESTÁVEL)	QUANDO TODAS AS ESCOLHAS DO PIVÔ RECAEM SOB A MEDIANA. $\theta(n \log n)$	QUANDO O PIVÔ É O MAIOR OU MENOR ELEMENTO DA LISTA $\theta(n^2)$	$\theta(n) \rightarrow$ Pior Caso $\theta(\log n) \rightarrow$ NO MELHOR CASO.	
PARTIÇÃO ( <u>NÃO</u> ESTÁVEL)	REQUER TEMPO $\theta(n)$		$O(1)$	
COUNTING SORT (ESTÁVEL)	A complexidade <u>temporal e espacial</u> é $\theta(n+k)$ . Se $k$ é $O(n)$ , tais complexidades serão $\theta(n)$ .			ORDENA LISTAS DE NÚMEROS NATURAIS.
BUCKET SORT (ESTÁVEL)	SE OS ELEMENTOS DE $L$ ESTIVEREM UNIFORMEMENTE DISTRIBUÍDOS NO INTERVALO $[0, k]$ O Bucket requer tempo <u>esperado</u> $\theta(n)$		$\theta(n)$	ORDENA LISTAS DE NÚMEROS NÃO NEGATIVOS.
RADIX SORT (ESTÁVEL)	tempo linear (Counting Sort ou Bucket Sort) $\theta(n \cdot k)$ se $k$ FOR CONSTANTE O RADIX SORT GASTARÁ TEMPO LINEAR $\theta(n)$		$\theta(n)$ SE USAR Bucket Sort ou Counting Sort.	