

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 (NÃO 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 (NÃO ESTÁVEL)	LISTA JÁ ORDENADA $\theta(n \log n)$	NÃO SE SABE, MAS ESTÁ ENTRE: $\theta(n \log n)$ e $\theta(n \log^2 n)$	$O(1)$	
MERGE SORT (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 $\theta(n)$			
QUICK SORT (NÃO 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 (NÃO 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.	