

UNIVERSIDADE DO ESTADO DE MINAS GERAIS – UEMG  
UNIDADE DIVINÓPOLIS  
Engenharia da Computação – 3º Período

## ANÁLISE DE EXECUÇÃO

Phelipe Gustavo da Silva

Divinópolis, MG  
Outubro 2016

Durante a implementação dos métodos de ordenação ShellSort e QuickSort foi possível perceber uma grande superioridade em relação aos métodos mais simples (BubbleSort, SelectionSort e InsertionSort) .

Resultados:

- Quantidade de passos

Passagens		
Lista Estática		
Método / Itens	500	2000
BubbleSort	62034	992597
SelectionSort	1891	7937
InsertionSort	62034	992597
ShellSort	2730	14913
QuickSort	<b>405</b>	<b>1503</b>

Passagens	
Lista Dinâmica	
Método / Itens	500
BubbleSort	62034
SelectionSort	1891
InsertionSort	62034
ShellSort	2730
QuickSort	<b>1067</b>

- Tempo em ms()

Tempo (ms)		
Lista Estática		
Método / Itens	500	2000
BubbleSort	2,068	24,50
SelectionSort	0,584	9,248
InsertionSort	0,502	6,257
ShellSort	0,093	0,410
QuickSort	<b>0,087</b>	<b>0,403</b>

Tempo (ms)	
Lista Dinâmica	
Método / Itens	500
BubbleSort	2,760
SelectionSort	0,880
InsertionSort	<b>0,680</b>
ShellSort	32,833
QuickSort	0,884

O método ShellSort utiliza princípios lógicos do InsertionSort, porém com algumas melhorias, no InsertionSort para a troca de um item na última posição era necessário comparar todos itens anteriores e realizar as trocas imediatamente, no ShellSort não é necessário essas trocas podendo realizar 'saltos', sua superioridade é visível no resultado final, a quantidade de passos e o tempo de execução foram reduzidos em mais da metade, porém sua implementação em listas dinâmicas é inviável obtendo um tempo maior até que o método BubbleSort.

Analisando o método QuickSort observamos uma superioridade sobre os outros métodos, incluindo o ShellSort, seu tempo de execução e número de passagens foram bem menores. O algoritmo utiliza de problemas matemáticos complexos e por isso existe uma grande dificuldade para sua implementação.

Por fim ressaltar a dificuldade da implementação dos métodos ShellSort e QuickSort em listas encadeadas, devido a falta de acessibilidade a um determinado item da lista, aumentando e muito o tempo de execução, como podemos ver o ShellSort com 500 itens levou 32,833 ms para a ordenação enquanto outros métodos levaram menos de 1ms.