



## Lista de Ordenação 2

### Bubble e Shell Sort

1. O desempenho da ordenação do Bubble Sort pode ser melhorado se pararmos o processo de classificação assim que descobriremos que em uma iteração, nenhuma troca de elementos ocorre. Escreva uma função em linguagem C que implemente o algoritmo usando esse fato.
2. Escreva uma versão do algoritmo em linguagem C do Bubble-Sort que faça a ordenação das strings na seguinte ordem: "Strings", "Shell\_Sort", "Ordenacao", "Bubble\_Sort". Imprima na tela o resultado.
3. Escreva um pseudocódigo para a versão recursiva do algoritmo de ordenação Bubble Sort.
4. Faça a ordenação por Shellsort para um vetor A com os seguintes valores:  $A < 12, 42, 1, 6, 56, 23, 52, 9, 5, 24 >$ . Considere gap igual a 3 para a resolução.
5. Escreva um programa em linguagem C que gere aleatoriamente números para um vetor contendo 100 números e após faça a ordenação em ordem decrescente por shell sort.
6. Crie um programa que cria dois vetores v1 e v2 não inicializados com 5000 elementos. Depois, preencha o v1 com valores aleatórios nas 5000 posições. Após, deve-se copiar v1 para v2 e ordenar com Bubble sort e Shell sort. Faça a medição do tempo de execução para cada método.  
Output esperado:  
"Tempo demorado com Bubble sort: x segundos"  
"Tempo demorado com Shell sort: y segundos"