UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
Bacharelado em Ciência da Computação e
Engenharia da Computação
INF 01124 – Classificação e Pesquisa de Dados

Ordenação Interna: ShellSort

Considere a implementação em C da função inserção direta *ShellSort* (método de incrementos decrescentes) para resolver os exercícios a seguir.

```
void main()
{
 int i , j , k, h = 1, value ;

   int vetor[] = {7, 2, 5, 6, 3, 1, 0, 8, 9, 4};
   int numero = 16;

   do { h = 3 * h + 1;  } while ( h < numero );

   do
   {   h /= 3;
       for ( i = h; i < numero; i++)
       {     value = vetor [ i ];
             j = i -h;
             while (j >= 0 && value < vetor [ j ])
             {     vetor [ j + h ] = vetor [ j ];
                   j=j-h;
             }
             vetor [ j + h ] = value;
       }

   } while ( h > 1 );
}
```

01. Utilize o algoritmo *ShellSort* para classificar os seguintes vetores:

```
vetor[ ] = {2, 6, 8, 4}
vetor[ ] = {2, 4, 6, 8}
vetor[ ] = {8, 6, 4, 2}
```

03. Qual o melhor incremento para o algoritmo *ShellSort*?

04. Faça uma breve comparação entre o caso médio e o pior caso para a execução do *ShellSort*.

05. O algoritmo *ShellSort* é **estável**? *Lembrando que um algoritmo de ordenação estável é aquele que não altera a ordem relativa das chaves iguais.*

( ) Sim ( ) Não

06. O algoritmo *ShellSort* é mais indicado para quais tipos de arquivo?

( ) Pequeno
( ) Moderado
( ) Grande

Por quê?

07. Podemos afirmar que a implementação do algoritmo *ShellSort* é simples e a implementação é pequena?

( ) Sim ( ) Não

Por quê?

08. Quais as vantagens e desvantagens do algoritmo *ShellSort*?

09. Em quais casos (aplicações) é indicado utilizar o algoritmo *ShellSort*? Justifique sua resposta.

10. Compare os algoritmos Inserção Direta, Inserção Direta com Busca Binária, *ShellSort* considerando:

- Melhor caso
- pior caso
- caso médio
- estável
- tamanho do arquivo de entrada
- implementação