

ALGORITMOS E ESTRUTURAS DE DADOS III TRABALHO PRÁTICO - ALGORITMOS DE ORDENAÇÃO Grupo: INDIVIDUAL Prazo de entrega: 16/05/2020

Modalidade: O trabalho deverá ser entregue em um único documento contendo os itens abaixo relacionados, e o código fonte deverá estar em anexo.

1) estudar o método Combsort, implementar, e dissertar seu funcionamento em um documento texto.

R: O CombSort é uma melhoria do método Bubble sort, e rivaliza com algoritmos como o Quicksort, ele tem o fator de redução que serve para reduzir o gap (tamanho de intervalo de troca) com valor de $gap = (int) \text{encolhimento} / 1.3$, e assim o código andar mais rápido, o fator é uma constante usada para dividir o gap.

```
void display(int *vetor, int tamanho)
{
    for(int i = 0; i < tamanho; i++)
        cout << vetor[i] << " ";
    cout << "\n";
}

void combSort(int *vetor, int tamanho)
{
    int gap = tamanho; //inicializa o tamanho do espaço com o tamanho da vetor
    bool flag = true;
    while(gap != 1 || flag == true)
    {
        gap = (gap*10)/13; //minimizar a folga pelo fator de contração
        if(gap < 1)
            gap = 1;
        flag = false;
        for(int i = 0; i < tamanho-gap; i++) //comparar elementos com gap
        {
            if(vetor[i] > vetor[i+gap])
            {
                swap(vetor[i], vetor[i+gap]);
                flag = true;
            }
        }
    }
}
```

2) utilizando o programa em C++ fornecido, o qual contém apenas o método Bolha implementado, implementar os demais métodos abaixo listados, executar o programa e apresentar os resultados de tempo gasto por cada algoritmo em um gráfico: a. Inserção Direta; b. ShellSort; c. BubbleSort; d. CombSort; e. QuickSort;

```

E:\16-02-20 Trabalho Algoritmo\Programa para Trabalho\Bolha.exe

===== ANALISANDO MÉTODOS =====
ORDENANDO VETOR DE 100000 VALORES

Método                Tempo
CombSort              0,01300 segundos
QuickSort             0,01700 segundos
ShellSort             0,01600 segundos
Inserção Direta       8,85000 segundos
BubbleSort            25,28700 segundos
Bolha                 22,20500 segundos

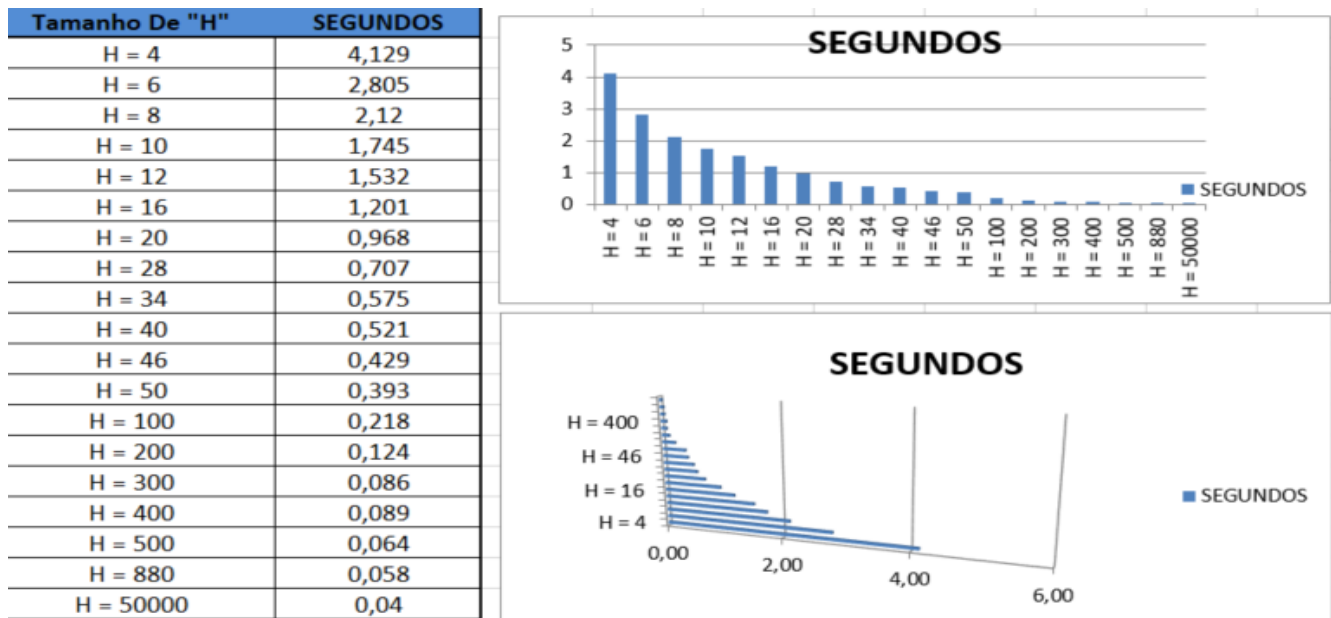
Pressione algo para encerrar...

```

3) considerando o método de classificação ShellSort, desenvolver um documento texto que comente, de forma simples, seu funcionamento e porque ele é um bom método de classificação. Efetuar também alguns testes a respeito do tamanho de h (ou número de passos), determinar um limite e gerar um gráfico no Excel ilustrando o progresso de h (ou número de passos) no eixo x em relação ao tempo gasto na classificação (eixo y).

R: O meto ShellSort Divide o vetor classificado em seguimentos a serem definidos, cada seguimento possui aproximadamente n/h chaves. Os seguimentos são separados por inserção direta após a classificação em seguida h é dividido por 2 e o vetor é reclassificado em cada seguimento h, até que esses processos seja 1.

Este é um bom método para ocasiões que temos um número de chaves pequeno ou já possuem alguma ordenação parcial, trazendo vantagem em cada passo fazendo classificações parciais do vetor e isso influencia nos passos seguintes.



4) Dissertar de uma forma simples, em um documento texto, como o método de classificação QuickSort trabalha.

R: O Quicksort é um dos métodos de ordenação dos mais eficientes, ele pertence a categoria de Ordenação por Troca utilizam o paradigma de programação Dividir para Conquistar. Dado a sequência de entrada, o método particionador em três seguimentos. A esquerda fica o S1 que são Elementos menores ou igual ao S2 “Chave particionadora”, e a direita fica o S2 elementos maiores ou igual a S2. A chave particionadora é escolhida e armazenada em uma variável temporária, essa chave vai comparando com os elementos da direita ou esquerda e assim recursivamente até não restarem seguimentos com comprimento iguais a 0 (zero) elementos.