Relatório do Trabalho de Ordenação

Nome: Gabriel Leite de Freitas
RA: 318122246

UNA – Barreiro

Introdução

O trabalho sugere cinco métodos de ordenação e com ele verificar o funcionamento e desempenho.

Bubble sort -
Neste algoritmo cada elemento da posição i será comparado com o elemento da posição i + 1, ou seja, um elemento da posição 2 será comparado com o elemento da posição 3. Caso o elemento da posição 2 for maior que o da posição 3, eles trocam de lugar e assim sucessivamente. Por causa dessa forma de execução, o vetor terá que ser percorrido quantas vezes que for necessária, tornando o algoritmo ineficiente para listas muito grandes.

Selection sort - Baseado em se passar sempre o menor valor do vetor para a primeira posição, depois o segundo menor valor para a segunda posição e assim sucessivamente.É repetido esse processo até que a lista esteja ordenada.

Insertion sort – Algoritmo simples e eficiente quando aplicado em pequenas listas. Percorre da esquerda para a direita, à medida que avança vai deixando os elementos mais à esquerda ordenados.

Quicksort –

Nele se escolhe um elemento chamado de pivô, a partir disto é organizada a lista para que todos os números anteriores a ele sejam menores que ele, e todos os números posteriores a ele sejam maiores que ele. Ao final desse processo o número pivô já está em sua posição final.

Merge Sort -

O vetor será dividido em duas partes iguais, que serão cada uma divididas em duas partes, e assim até ficar um ou dois elementos cuja ordenação é trivial. Para juntar as partes ordenadas os dois elementos de cada parte são separados e o menor deles é selecionado e retirado de sua parte. Em seguida os menores entre os restantes são comparados e assim se prossegue até juntar as partes.

Explicação

Para executar o trabalho moldei uma função para cara criação dos vetores e uma função para cada método de ordenação.

Usando o programa MAIN desenvolvi vetores para receber os resultados das funções e apliquei as cinco funções tema do trabalho.

Para calcular o tempo adicionei a biblioteca using System Diagnostic para utilizar o comando StopWatch(), o tal funciona de seguinte forma, uma variável para receber o valor inicial, final e o tempo de execução.

- Variavel.start (inicia o tempo de execução)
- Variavel.stop (finaliza o tempo de execução)
- Variavel.elapsed (Mostra o tempo de execução do método a ser medido)

Ao final coloquei cada vetor (crescente, decrescente e randomico) em cada método de ordenação, adicionei o comando.Elapsed para demonstrar o tempo de execução em todas as situações. E gerei os números randomicos utilizei o comando Random().

Analise da execução

Infelizmente não consegui organizar o tempo para tirar os prints de tela, porém testei todo o código que está no GitHub e está funcionando normalmente.

Tempo de execução Insertion sort

Tempo de execução Selection sort

Tempo de execução Bubble sort

Tempo de execução Quick sort

Tempo de execução Merge sort

Conclusão

Durante o trabalho consegui aprender e diferenciar os casos para cada método de ordenação. O que mais me chamou a atenção foi a utilização do Stopwatch, que seria como se fosse um cronômetro para conseguir buscar o tempo, não conhecia essa opção até dar uma pesquisada e fazer alguns testes também para ver seu funcionamento, realmente é muito interessante e auxilia muito para se organizar bem com os tempos de execução dos processos.

O trabalho como uma forma geral foi muito interessante, realmente não estava entendendo muito bem o C# como um todo no início do semestre, mas com o desenvolver do trabalho pude ir me aprimorando conhecendo novas ferramentas, testando com calma e a cada dia fazendo um pouco, assim praticando mais e absorvendo mais fácil as informações.