

Comentários:

1. Funções para Geração dos Números Aleatórios, Gerenciamento dos Arquivos e Aferição do Tempo:

Todas as funções necessárias que são precedentes à execução dos programas “inverteOrdem.c” e “bubble.c” foram desenvolvidas com inspiração naquelas elaboradas pelo professor em aula, mas com algumas adaptações, que julguei convenientes para otimização dos testes.

2. A Função Main:

A função principal de ambos os programas é bastante similar. A única diferença se dá na função central a ser executada. O tempo de execução da função main, sem levar em consideração a função central do programa, aumenta linearmente de maneira proporcional ao número de execuções (executionsNumber) passado pela linha de comando.

3. Inversor de Ordem:

Na primeira linha da função, existem 2 atribuições, e 1 operação aritmética. A inicialização no laço “for” contém 2 atribuições e 1 operação aritmética. Sua condição de parada tem 1 comparação e seu incremento tem duas atribuições e duas operações aritméticas. Dentro do seu escopo existem 3 atribuições e 4 acessos a ponteiro.

Na primeira chamada do laço for, ocorre a inicialização e a condição de parada. A partir daí, são executados o bloco de comandos do seu escopo, a condição de parada e o incremento, todos repetidos $(n / 2)$ vezes, em que $(n / 2)$ equivale à variável halfSize.

Portanto, o total de operações, já na forma simplificada, equivale à $(n / 2) (5a + 4p + 2o + c) + 2(2a + o) + c$, em que n = número de elementos da lista, a = atribuição, p = acesso a ponteiro, o = operação aritmética e c = comparação. A expectativa de padrão de crescimento linear é corroborada pelo gráfico.

4. Ordenador Bolha:

A inicialização, condição de parada e incremento de cada laço de repetição “for” é dada por uma atribuição, uma comparação e uma atribuição e operação aritmética, respectivamente. Na primeira linha do escopo do segundo laço “for” há 2 atribuições, 2 acessos a ponteiro e uma operação aritmética. Considerando o pior caso, existirão ainda uma comparação, 2 atribuições, 2 acessos a ponteiro e uma operação aritmética.

Na primeira execução do primeiro laço “for”, serão executadas a inicialização e a condição de parada. A partir daí, o segundo laço “for”, a condição de parada e o incremento são todos repetidos n vezes, em que n equivale à variável “numberList->size”.

Na primeira execução do segundo laço “for”, serão executadas a inicialização e a condição de parada. A partir daí, o bloco de comandos no seu escopo, a condição de parada e o incremento são todos repetidos $(n / 2)$ vezes, em que $(n / 2)$ equivale à metade da variável “numberList->size”.

Portanto, o total de operações, já na forma simplificada, equivale à $(n^2 / 2) (5a + 4p + 3o + 2c) + n (2a + 2c + o) + a + c$, em que n = número de elementos da lista, a = atribuição, p = acesso a ponteiro, o = operação aritmética e c = comparação. A expectativa de padrão de crescimento quadrático é também corroborada pelo gráfico.