

Trabalho de Implementação 1

Os algoritmos devem ser implementados em C ou C++. As entradas e saídas devem ser feitas por arquivo texto. O uso de interface gráfica é opcional (ou seja, não é necessário).

1. Implemente os algoritmos MergeSort, HeapSort e QuickSort (com pivô aleatório). Gere vetores aleatórios de tamanhos 1000, 2000, 3000 até 100.000 de números entre 0 e 1, execute cada algoritmo de ordenação e contabilize o tempo de cada. Gere uma curva relacionando os tempos com os tamanhos dos vetores.

2. Seja $P : \mathbb{N} \rightarrow \mathbb{N}$ uma função definida da seguinte forma: $P(0) = P(1) = P(2) = P(3) = P(4) = 0$ e, para $n \geq 5$,

$$P(n) = P\left(\lfloor \frac{n}{2} \rfloor\right) + P\left(\lfloor \frac{n}{2} \rfloor + 1\right) + P\left(\lfloor \frac{n}{2} \rfloor + 2\right) + n.$$

Implemente três algoritmos (um recursivo puro, um de memoização e um de programação dinâmica) que recebem um número n como entrada e retorna o valor exato de $P(n)$. Para cada valor de n variando de 0 até 1.000.000, calcule $P(n)$ usando cada algoritmo, contabilize o tempo de cada e faça uma curva relacionando os tempos com os valores de n .

3. Implemente um algoritmo de memoização e um de programação dinâmica para o problema da multiplicação de seqüências de matrizes. Gere vetores aleatórios de inteiros (representando as dimensões das matrizes) com tamanhos 1000, 2000, 3000 até 100.000, execute cada algoritmo e contabilize o tempo de cada. Gere uma curva relacionando os tempos com os tamanhos dos vetores.