

# Programação Concorrente e Paralela



## Mini EP 3

**Aluno(s):**

Thales V. S. Lobo de Almeida [16311010]

**Professor:**

Alfredo Goldman Vel Lejbman

## 1 Desempenho Inicial

O *profiling* inicial indicou que as funções `slowsort` e `fibonacci` eram responsáveis pela maior parte do tempo de execução, consumindo 54,17% e 43,05%, respectivamente. Além disso, o *profiling* revelou uma quantidade elevada de chamadas recursivas e chamadas no geral, especialmente na função `fibonacci` (cerca de 200 chamadas recursivas), que recalculava valores já processados repetidas vezes. Esses fatores evidenciaram a necessidade de otimizações nessas partes críticas do código.

## 2 Otimizações Implementadas

Para corrigir os problemas de desempenho, foram aplicadas as seguintes ideias de otimizações:

1. **Substituição do Slowsort pelo Quicksort:** O algoritmo `slowsort`, conhecido por sua ineficiência com complexidade  $O(n^2)$ , foi substituído pelo `quicksort`, que possui complexidade  $O(n \log n)$ .
2. **Memoization no Cálculo de Fibonacci:** A função de cálculo de Fibonacci foi modificada para utilizar memoization. Em vez de recalcular valores repetidamente, os resultados já computados são armazenados e reutilizados, reduzindo drasticamente o número de chamadas recursivas.
3. **Pré-cálculo de Fibonacci:** Além da memoization, os números de Fibonacci até o maior valor presente no vetor foram pré-calculados. Essa estratégia eliminou a necessidade de cálculos repetitivos durante a fase de soma dos valores, além do cálculo de valores não desejados (valores ímpares).

## 3 Resultados do *profiling* Após as Otimizações

O *profiling* realizado após as otimizações mostrou uma redução significativa no tempo de execução, tornando-o negligível nas métricas obtidas (os tempos de execução e processamento totais foram todos inferiores a escala utilizada). As funções anteriormente problemáticas, como `slowsort` e `fibonacci`, não foram mais identificadas como gargalos, indicando que as otimizações surtiram o efeito desejado.