

## COMPARAÇÃO DO TEMPO DE EXECUÇÃO DE ALGORITMOS DE BUSCA DE MAIOR E MENOR ELEMENTO EM PROCESSADORES DIFERENTES

Roberto G. D. de Oliveira<sup>1</sup>, robertoguimaraes8@hotmail.com; Renner Augusto A Lima<sup>1</sup>, renneramil@gmail.com; Márcio O. Costa<sup>2</sup>, costafilo@yahoo.com.br; Walteno M. Parreira Júnior<sup>3</sup>, waltenomartins@yahoo.com.

<sup>1</sup>Graduandos do Curso de Engenharia de Computação – UEMG-FEIT;

<sup>2</sup>Professor do curso de Engenharia da Computação – UEMG-FEIT;

<sup>3</sup>Professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistema de Informação - UEMG-FEIT.

Este trabalho relata uma das atividades desenvolvidas no projeto de estudo e implementação de algoritmos de busca de menor e maior elemento de um vetor, testando sua eficiência em máquinas distintas. A idéia principal foi realizar a comparação entre os tempos encontrados na execução de algoritmos em processadores diferentes. Segundo Terada, um algoritmo é, em geral, uma descrição passo a passo de como um problema é solucionável. A descrição deve ser finita, e os passos devem ser bem definidos, sem ambigüidades, e executáveis computacionalmente. Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa. Os algoritmos foram implementados na Linguagem C, compilados no Borland C++ versão 5.02 e executados em três computadores com especificações diferentes, permitindo que seus tempos de execução fossem comparados. Os processadores utilizados foram: AMD Turion 64X2 TL-50 1GB RAM DDR2 2x256kb Cache. O segundo computador AMD Athlon XP 1,49GHz 1GB RAM 256kb cachê. E o terceiro Intel Core 2 duo T5550 1,83GHz 4GB RAM 2MB cachê. Os algoritmos implementados fazem parte do método divisão e conquista. Segundo Parreira Júnior, tal método diz que dado um problema, de tamanho  $n$ , o método divide-o em duas ou mais partes, criando subproblemas menores. Cada subproblema é resolvido separadamente e então as soluções são combinadas para a obtenção da solução da instância original. Foram implementados e testados três algoritmos. O Maxmin1 é o código mais simples dos três, mas também é o código menos sofisticado e menos eficiente. Em todos os casos possíveis (melhor, pior e caso médio) são realizadas  $2(n-1)$  processos. O Maxmin2 possui uma lógica que reduz a quantidade de processos e também o tempo de execução, onde o pior caso do Maxmin2,  $2(n-1)$ , será igual ao melhor caso do Maxmin1. O Maxmin3 é um código mais refinado e eficiente do que os dois primeiros e realiza a mesma quantidade de processos em todos os casos. Após a execução têm-se que o AMD Athlon, que é o processador mais lento dos três, foram encontrados tempos menores do que o AMD Turion que é um processador um pouco mais rápido. Isso ocorreu para os três algoritmos. Isso acontece por que o processador não é o único fator que interfere diretamente na velocidade de execução de um programa, a memória RAM também tem impacto direto sobre a velocidade de execução. Já o Intel Core 2 duo, que possui um processador mais rápido e 4GB de RAM, superou os outros dois processadores e obteve tempos de execução menores em todos os três algoritmos. Outro fator é que os compiladores mais antigos não foram desenvolvidos para trabalharem com mais de um núcleo ao mesmo tempo. Concluímos que para desenvolver programas é necessário entender a arquitetura dos computadores e entender quais os fatores influenciam na velocidade de execução de programas. Também é importante entender como trabalham os programas em cima de processadores de múltiplos núcleos.

Área: Ciências Exatas e dos Materiais

**Para referenciar:**

OLIVEIRA, R. G. D. de, LIMA, R. A. A., COSTA, M. O., PARREIRA JÚNIOR, W. M. Comparação do Tempo de Execução de Algoritmos de Busca de Maior e Menor Elemento em Processadores Diferentes In: II Mostra e II seminário de Ensino, Pesquisa e Extensão da FEIT, 2010, Ituiutaba. **Anais**. Ituiutaba: FEIT, 2010.