

Durante as últimas décadas foram expressivos os avanços dos microprocessadores, no entanto a velocidade das memórias não acompanhou esse desenvolvimento. Portanto, se desenvolvermos algoritmos que minimizem a necessidade do processador de procurar dados na memória iremos acelerar o seu processamento. Os algoritmos desenvolvidos com esse propósito são conhecidos como algoritmos cache-oblivious.

Com o objetivo de estudar essa área de análise e desenvolvimento de algoritmos, implementamos e analisamos o desempenho do algoritmo cache-oblivious, proposto por Chowdhury e Ramachandran, que resolve o problema da maior subsequência comum. Dadas duas sequências $X = (x_1, \dots, x_n)$ e $Y = (y_1, \dots, y_n)$, nós queremos encontrar a maior subsequência $Z = (z_1, \dots, z_k)$ comum a X e a Y . Esse problema possui importantes aplicações na biologia, como na comparação entre genomas de diversos seres vivos.

A partir dessa análise constatamos que o algoritmo cache-oblivious é 120% mais rápido quando comparado com o algoritmo tradicional. Isso porque ele divide o problema em sub-problemas menores recursivamente até que os dados caibam inteiramente na memória cache. Essa divisão favorece a localidade espacial e temporal dos acessos aos dados, o que diminui os acessos a memória e aumenta a velocidade de execução do algoritmo.