



Faça o que é pedido e poste no SGA até **dia 18/11/2019 às 23:59h**.

MergeSort e o Quicksort paralelizados

Implemente versões sequenciais e paralelizadas do MergeSort e do QuickSort a fim de coletar tempos de execução e calcular o *speedup*. Sempre execute cada versão pelo menos 10 vezes e reporte valores médios.

- a) Uma das versões do Mergesort vai fazendo chamadas recursivas no código paralelizado e diminuindo um "contador" de chamadas recursivas pela metade (stop/2). Quando o contador chega em 1, o código executado não é mais o paralelo, mas o sequencial. Isso faz sentido, já que a cada chamada recursiva o Mergesort "divide" o vetor que está sendo ordenado por dois. Ou seja, quando o vetor é "pequeno" não compensa paralelizar, pois a sobrecarga (*overhead*) das *threads* faz o processamento demorar mais. Para testar isso, em vez de dividir o contador de *threads* por dois, faça stop – 1 e, depois, stop – 2... nas chamadas recursivas paralelizadas.
- b) Altere o parâmetro de `omp_set_nested()` e repita o que foi feito no Item a).
- c) Altere o número de *threads* para potências de 2: 2, 4, 8, 16, 32, 64... `MAX_THREADS`. Para cada valor rode todas as versões a fim de determinar qual o melhor número de *threads*. Você define o valor de `MAX_THREADS`.
- d) Monte tabelas com os resultados para cada algoritmo (Quick e Merge) e para cada versão e escreva um parágrafo comentando os resultados e destacando os melhores. Para que os resultados sejam comparáveis sem que a ordem inicial dos elementos do vetor influencie, fixe um tamanho de vetor, gere valores randômicos e faça uma cópia do vetor. A cada execução use a cópia do mesmo vetor desordenado para ser entrada dos algoritmos.

Um código do QuickSort, ainda a ser mais bem organizado sem o cálculo das médias e *speedup*, sem as diretivas de paralelização e sem a lógica completa para interromper a paralelização, será postado no SGA. Use-o como ponto de partida inclusive para o MergeSort.

Poste no SGA, até a data e horários especificados aqui, os resultados e todo código que produziu em um único arquivo texto.