

Atividade 04 ... classificação com threads e OpenMP

Implemente os algoritmos de classificação abaixo, executando-os nos formatos *monobloco* (um só processo), *multithread* (utilizando 2 ou mais threads) e OpenMP (dividindo tarefas, se possível!).

Monte arquivos de números inteiros com 2^{10} , 2^{14} , 2^{18} , 2^{22} , 2^{26} e 2^{30} elementos, gerados aleatoriamente, para testar cada implementação.

Após os testes com os seis arquivos, monte uma tabela comparativa do número de operações empregadas em cada algoritmo e, do tempo de processamento em cada formato.

Algoritmos de classificação

ShakeSort(A, n)

```
1 e ← 1
2 para i ← n-1 até e faça
3   para j ← e até i faça
4     se A[j] > A[j+1]
5       então troca(A[j], A[j+1])
6   para j ← i até e + 1 faça
7     se A[j-1] > A[j]
8       então troca(A[j-1], A[j])
9 e ← e + 1
```

Mergesort(A, e, d)

```
1 se e < d
2   então m ← ⌊ (e + d)/2 ⌋
3   Mergesort(A, e, m)
4   Mergesort(A, m+1, d)
5   Intercala(A, e, m, d)
```

Intercala (A, e, m, d)

```
0 crie vetor B[e..d]
1 para i ← e até m faça
2   B[i] ← A[i]
3 para j ← m+1 até d faça
4   B[d+m+1-j] ← A[j]
5 i ← e
6 j ← d
7 para k ← e até d faça
8   se B[i] ≤ B[j]
9     então A[k] ← B[i]
10    i ← i+1
11   senão A[k] ← B[j]
12    j ← j-1
```
