

Dica 1

Radix Sort

Os algoritmos de *radixsort* podem ser classificados em dois grupos: *least significant digit* (LSD) e *most significant digit* (MSD).

Os algoritmos LSD analisam os dígitos da chave da direita para a esquerda, trabalhando com os dígitos menos significativos. Já os algoritmos MSD avaliam os dígitos da esquerda para a direita (consequentemente, trabalhando com os dígitos mais significativos).

A Figura 1 apresenta um exemplo com os algoritmos LSD e MSD com chaves de tamanho 3.

362	291	207	207		237	237	216	211
436	362	436	253		318	216	211	216
291	253	253	291		216	211	237	237
487	436	362	362		462	268	268	268
207	487	487	397		211	318	318	318
253	207	291	436		268	462	462	460
397	397	397	487		460	460	460	462
LSD Radix Sorting				MSD Radix Sorting				

Figura 1. Exemplo simples dos algoritmos LSD e MSD.

Observando com calma o algoritmo MSD, é possível perceber que, após a 1ª ordenação – 2ª coluna da Figura 1 – todas as chaves iniciadas pelo mesmo dígito foram agrupadas.

Um exemplo pode ser visto na Figura 2.

237	237
318	216
216	211
462	268
211	318
268	462
460	460

Figura 2. Agrupamento de mesmo dígito – 1ª ordenação.

Para o problema solicitado na Maratona de Programação Paralela ERAD-SP/2012, seriam 93 grupos de chaves com o 1º dígito já ordenado – chaves agrupadas desde o ASCII 0x21 até 0x7E.

Portanto, a partir dessa situação, é possível pensar em uma solução paralela e/ou distribuída que considera cada grupo resultante desse 1º passo da ordenação como sendo tarefas – cada uma destas, independente. A Figura 3 apresenta um exemplo.

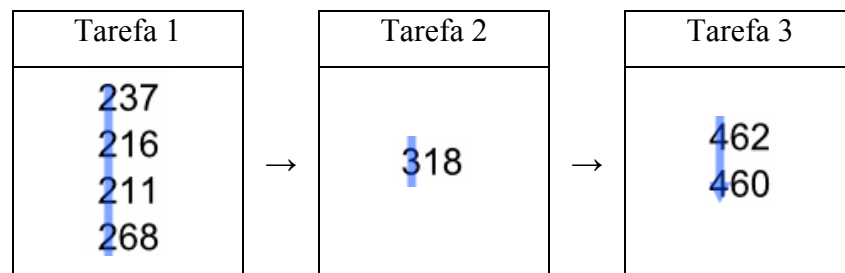


Figura 3. Tarefas geradas a partir do 1º passo da ordenação.

Sendo assim, é possível executar o 2º passo do algoritmo MSD (ou LSD) nos dígitos restantes de forma que cada processador execute o restante em paralelo.