Ordenação de Dados

Ordenação de dados

Interna

- Em memória principal

Externa

- Em memória secundária

Ordenação externa

Dados a serem ordenados não cabem na memória principal ou virtual (ex. atual: *Big Data*)

Algoritmos apropriados à latência de acesso da memória secundária (ou outras restrições semelhantes)

- Read-write lento
- Padrões de acesso restrito (ex. Fita - sequencial)

Algoritmos adequados devem levar em conta o dispositivo de armazenamento

Ordenação externa

Abordagem geral:

- Minimizar I/O lendo-escrevendo grandes blocos
- Acessos sequenciais terão melhor desempenho
- Levar em conta a hierarquia de memória (principal, caches, virtual)

Abordagem geral

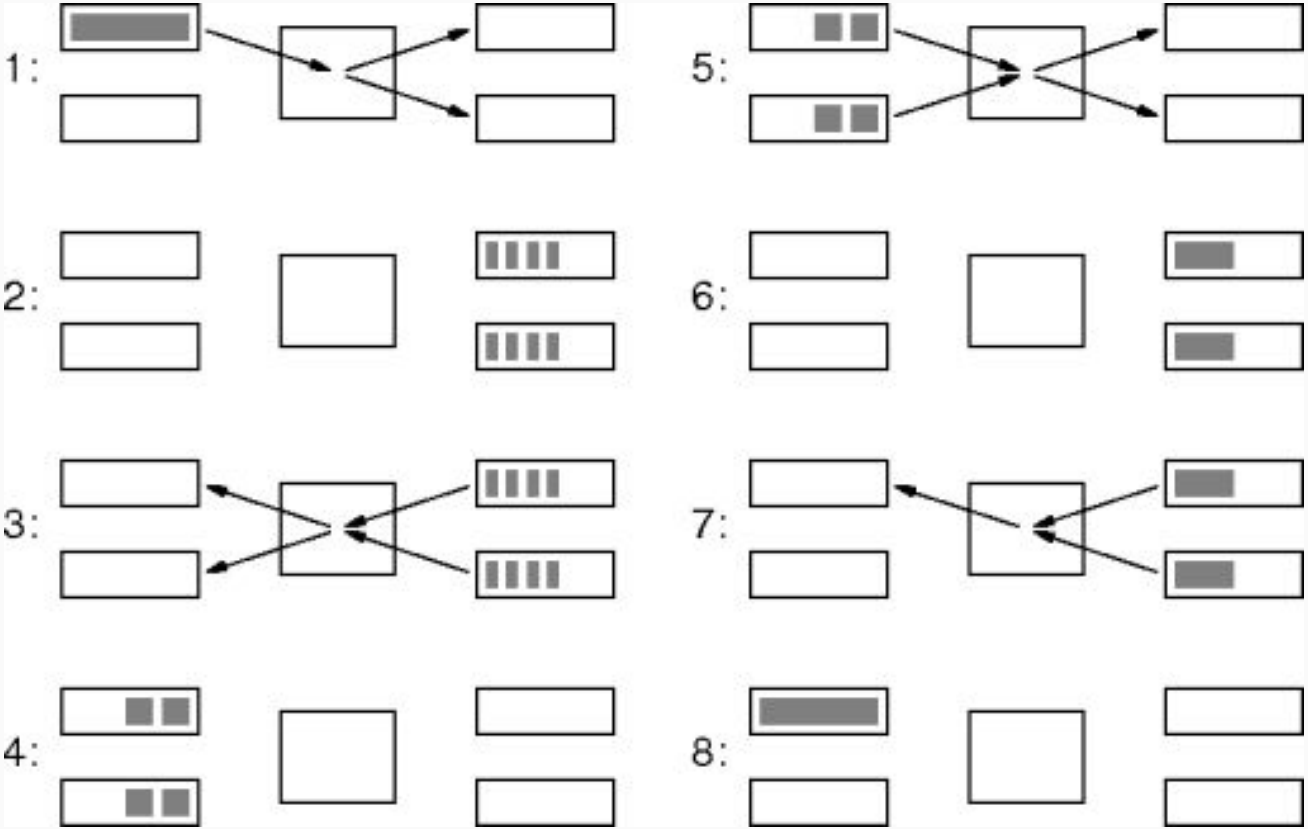
Merge sort (ou variações)

K-way merge

Merge sort balanceado de n-vias

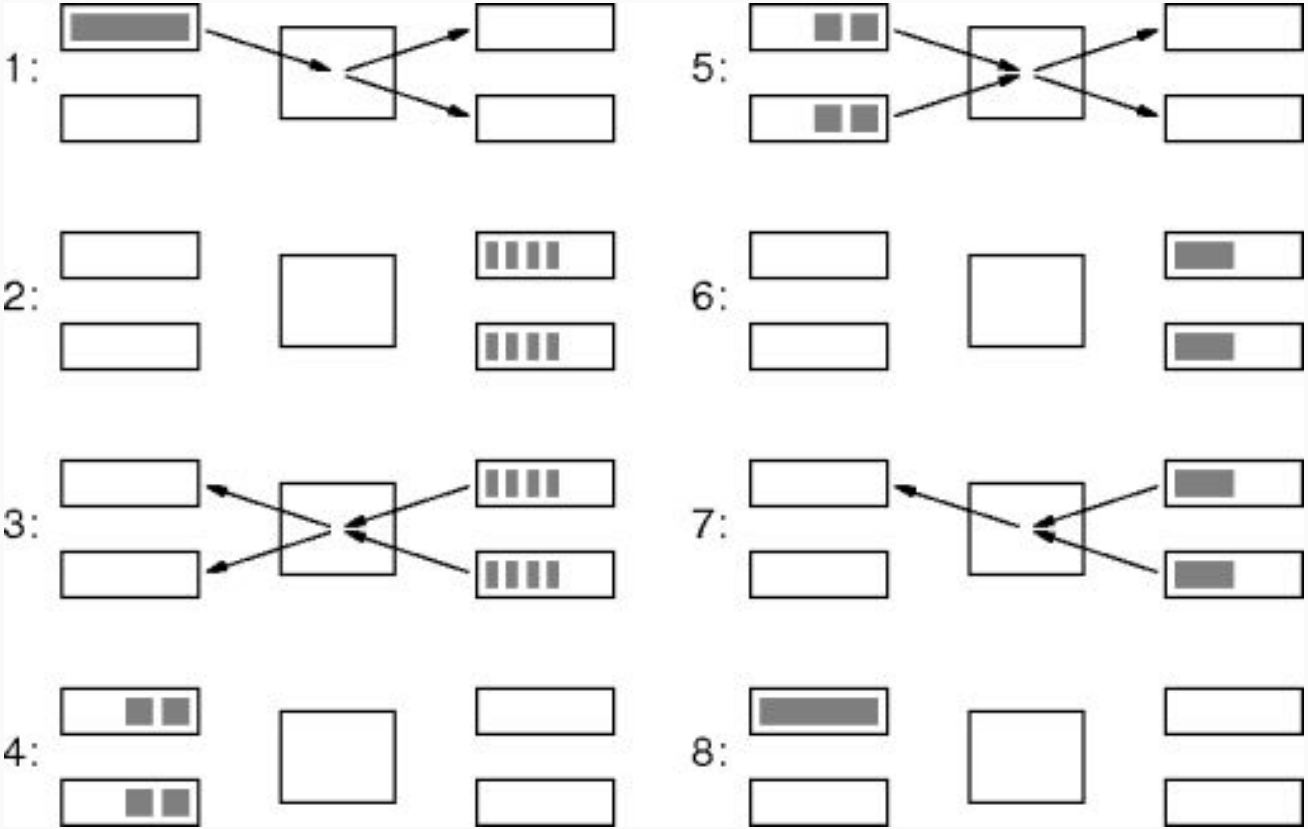
- Nesta abordagem, inicialmente define-se quantas entradas serão usadas

Merge sort balanceado de n-vias (ex. 2 entradas, 2 saídas)



Merge sort balanceado de n-vias (ex. 2 entradas, 2 saídas)

Problema: desperdício de dispositivos



Merge Sort Polifásico (desbalanceado)

1) Distribuir arquivos de entrada nos dispositivos, exceto um, que será a saída
2) Fazer *merge* das entradas na saída, até alguma das entradas esvaziar
3) Entrada vazia se torna nova saída, antiga saída se torna uma entrada
4) Enquanto dados não-ordenados, voltar para (2)

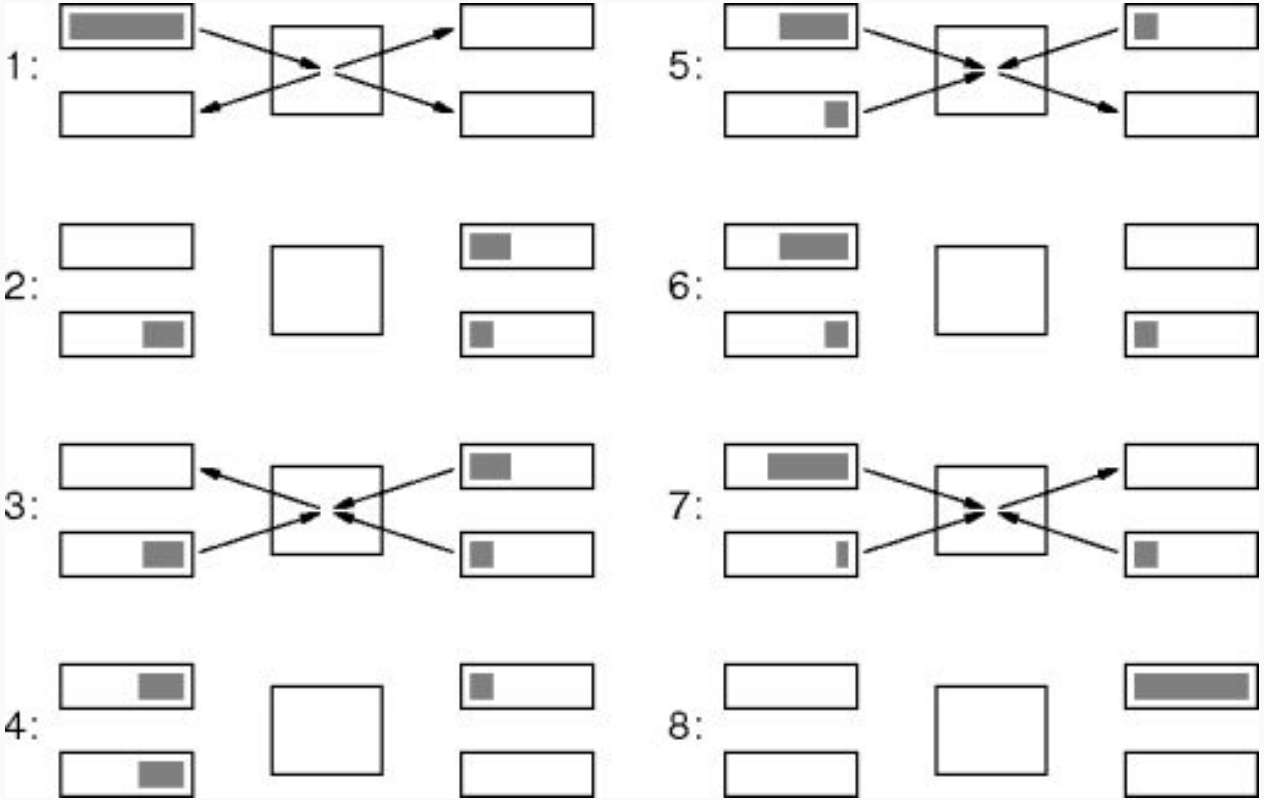
OBS: distribuição inicial deve ser não-uniforme (em tamanho de blocos e distribuição)

Merge Sort Polifásico

D_1	D_2	D_3	D_4
0	13	11	7
7	6	4	0
3	2	0	4
1	0	2	2
0	1	1	1
1	0	0	0

- Exemplo com 4 dispositivos
- Estes números são conhecidos como generalizações de Fibonacci
- Se necessário, são usados *Dummy blocks* para atingir estes números

Merge sort polifásico (desbalanceado)



External distribution Sort

- Similar a quicksort
  - Encontrar M/B pivôs (M = memória principal, B = Bloco que cabe na memória)
  - Usar os pivôs para dividir os N elementos em grupos, tal que cada um tenha elementos menores que o seguinte
  - Trazer um grupo para a memória e aplicar recursivamente até que o tamanho destes subgrupos seja tal que caibam na cache. Ordenar o grupo e gravá-lo em ordem na saída. Repetir para todos os N grupos