Tabela de dispersão

index = código ascii % nº de elementos do array

fator de carga = $\frac{\text{nº total de items}}{\text{tam do array}}$

Resolução de colisões:

- Linear probing → para a frente
- Quadratic probing → $H+i^2$ $(H = m\%k)$
- Separate chaining → caixas

Linear probing

679 16 68 42 38 116

679 → 22 % 11 = 0

16 → 7 % 11 = 7

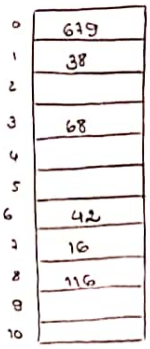
68 → 14 % 11 = 3

42 → 6 % 11 = 6

38 → 11 % 11 = 0 → 1

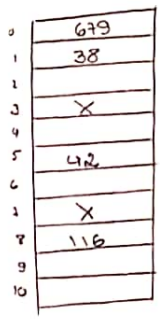
116 → 8 % 11 = 8

dimensão da matriz: 11



Eliminar 16 e 68 (Pag34)

com soma dos algarismos



Quadratic probing

679 16 68 42 38 116

dimensão da matriz: 13

679 % 13 = 3 → $3+0^2 = 3$

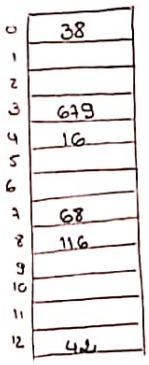
16 % 13 = 3 → $3+1^2 = 4$

68 % 13 = 3 → $3+2^2 = 7$

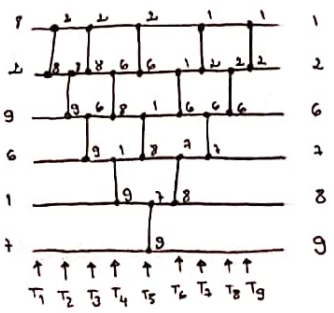
42 % 13 = 3 → $3+3^2 = 12$

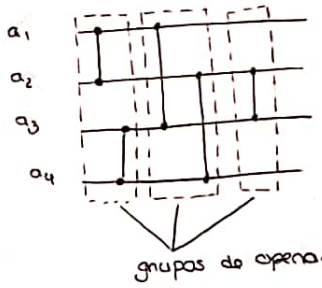
38 % 13 = 12 → $12+1^2 = 13 \to 0$

116 % 13 = 12 → $12+3^2 = 21 \to 8$



Redes de ordenamento





→ profundidade da rede = 3
↳ nº de operações que não podem ser paralelizáveis

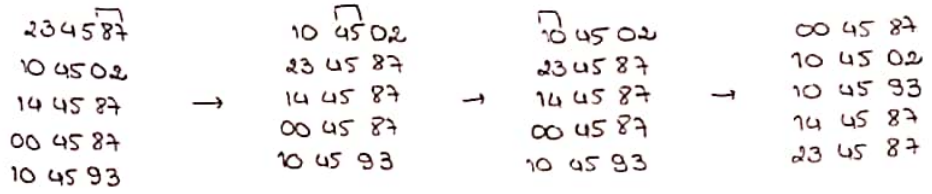
grupos de operações em paralelo

Algoritmo: Bubble sort, pois compara os algoritmos 2 a 2 e no fim de uma ronda já está ordenado.

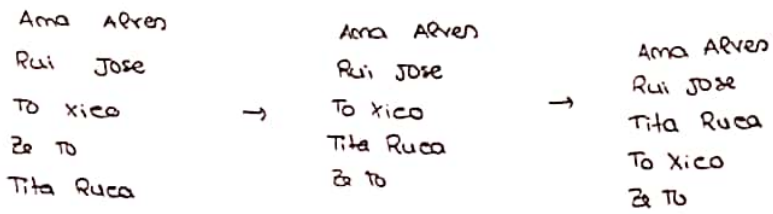
Redes ordenadas → ordenada (ex.: 0011)

Redes bitónicas → não ordenada ou todos os números iguais

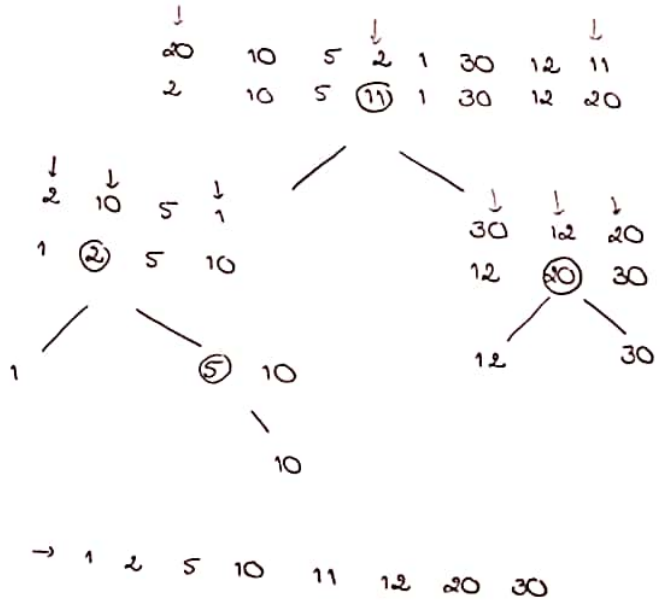
LSD



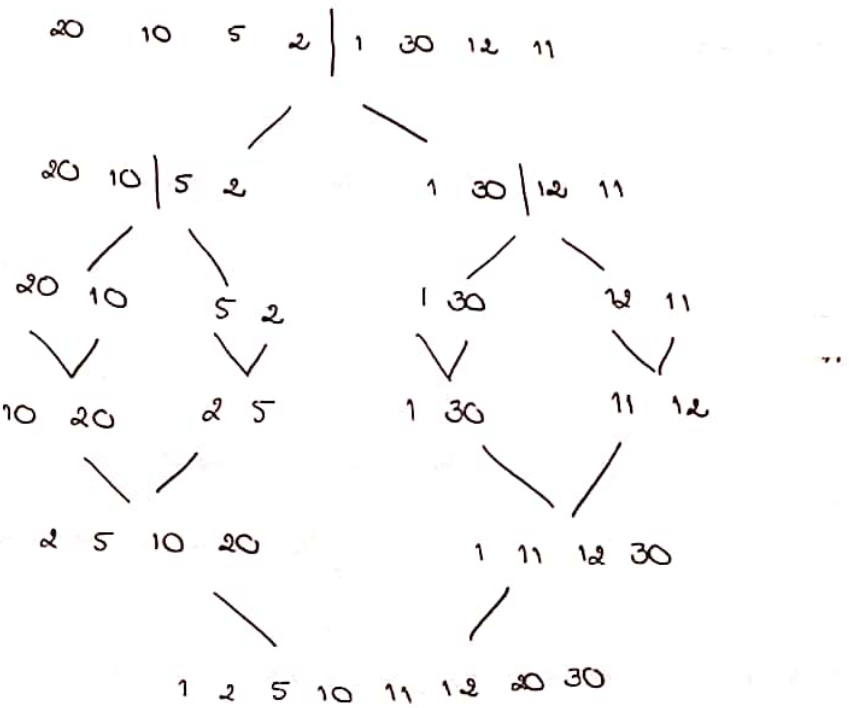
MSD



Quick sort

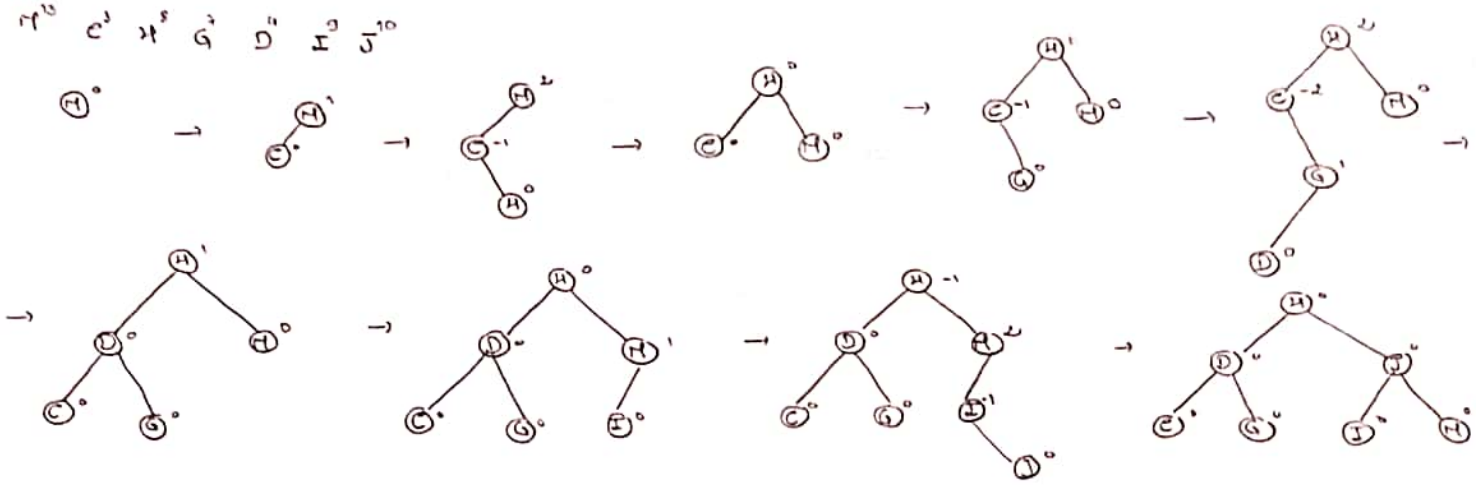


Menge sort



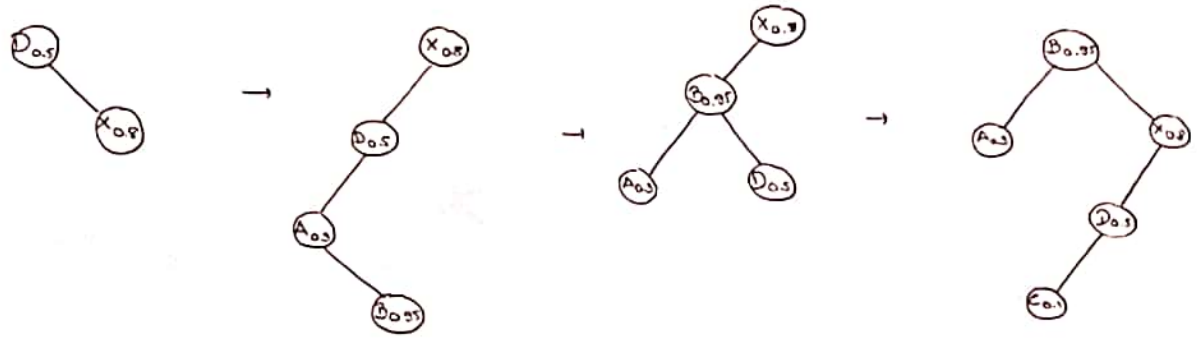
AVL

M[10] C[1] H[3] G[2] D[4] I[3] J[10]



Treap Max

D0.5 X0.8 A0.3 B0.95 Z0.2 C0.1



B-tree

→ cada nó tem m-1 chaves
→ até m filhos

Bloco em disco: 4096 bytes
Nó: 64 bytes
Referência: 4 bytes

64 + 4 = 68 bytes

4096 : 68 = 60 bytes → chaves de cada nó

m - 1 = 60 ⇔ m = 61 bytes

Treap

→ árvore binária de pesquisa
→ com prioridades
→ O(log m)

Heap sort

→ árvore binária
→ nenhum nó tem valor inferior ao dos seus descendentes
→ equilibrada
→ complexidade O(log m)

Radix sort

→ complexidade O(km)

· LSD → da dir para a esq
· MSD → da esq para a dir

Programação dinâmica → armazenamento de cálculos repetidos, permitindo a sua reutilização em chamadas recursivas idênticas

AVL

→ árvore binária de pesquisa

→ equilibrada

→ rápida operação de consulta

VP

→ árvore binária de pesquisa

→ cada nó é vermelho ou preto

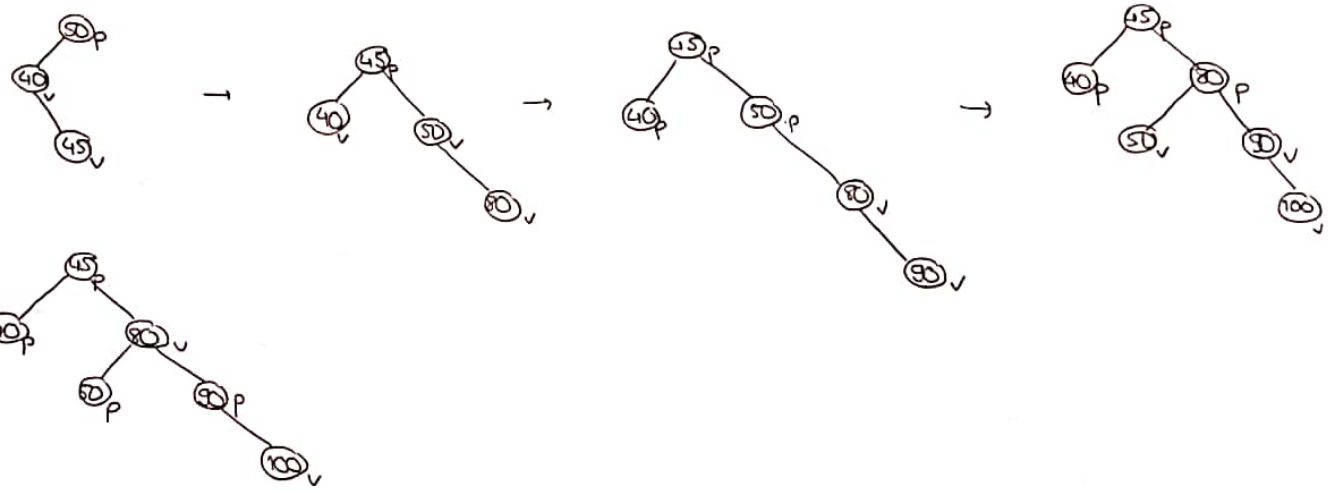
→ a raiz é preta

→ se um nó é vermelho os seus filhos são pretos

→ todos os caminhos raiz-folha têm o mesmo número de nós pretos

VP

50 40 45 80 90 100



20 10 5 3 7 1 4 2