

RADIX SORT

O Radix sort é um algoritmo de ordenação rápido e estável que pode ser usado para ordenar itens que estão identificados por chaves únicas. Cada chave é uma cadeia de caracteres ou número, e o radix sort ordena estas chaves em qualquer ordem relacionada com a lexicografia.

Na ciência da computação, radix sort é um algoritmo de ordenação que ordena inteiros processando dígitos individuais. Como os inteiros podem representar strings compostas de caracteres (como nomes ou datas) e pontos flutuantes especialmente formatados, radix sort não é limitado somente a inteiros.

1954



Cartões Perfurados



Harold H. Seward

Existem duas classificações do radix sort, que são:

- ▣ Least significant digit (LSD – Dígitos menos significativos).
- ▣ Most significant digit (MSD – Dígitos mais significativos).

LSD – Dígitos menos significativo

O radix sort LSD começa do dígito menos significativo até o mais significativo, ordenando tipicamente da seguinte forma: chaves curtas vem antes de chaves longas, e chaves de mesmo tamanho são ordenadas lexicograficamente. Isso coincide com a ordem normal de representação dos inteiros, como a sequência "1, 2, 3, 4, 5, 6, 7, 8, 9, 10".

Exemplo

329		720		720		329
457		355		329		355
657		436		436		436
839>	457>	839>	457
436		657		355		657
720		329		457		720
355		839		657		839

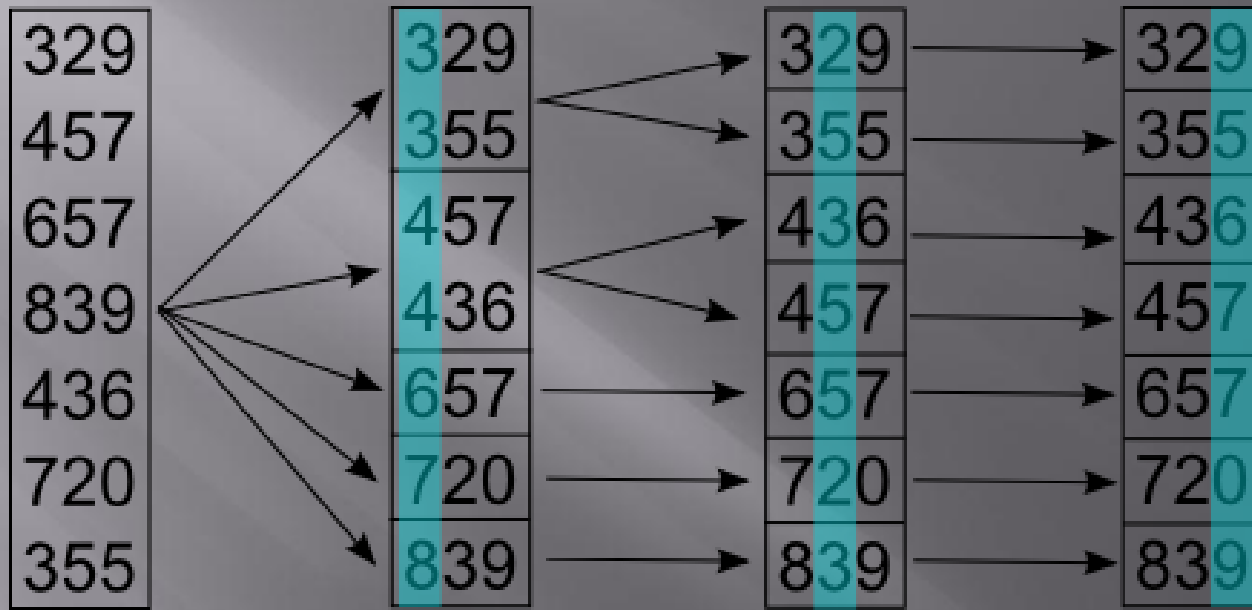
Passo a Passo

- Pegue o dígito menos significativo de cada chave.
- Agrupe as chaves com base nesse dígito, mas mantenha a ordem original das teclas.
- Repita o processo de agrupamento com cada dígito mais significativo.

MSD – Dígitto mais significativo

Já o radix sort MSD trabalha no sentido contrário, usando sempre a ordem lexicográfica, que é adequada para ordenação de strings, como palavras, ou representações de inteiros com tamanho fixo. A sequência "b, c, d, e, f, g, h, i, j, ba" será ordenada lexicograficamente como "b, ba, c, d, e, f, g, h, i, j". Se essa ordenação for usada para ordenar representações de inteiros com tamanho variável, então a representação dos números inteiros de 1 a 10 terá a saída "1, 10, 2, 3, 4, 5, 6, 7, 8, 9".

Exemplo



Passo a Passo

- Pegue o dígito mais significativo de cada tecla.
- Classifique a lista de elementos com base nesse dígito, agrupando elementos com o mesmo dígito em um balde.
- Ordene recursivamente cada um, começando com o próximo dígito à direita.
- Concatenar os baldes juntos em ordem.

Velocidade e Tamanho

Em muitas aplicações em que é necessário velocidade, o radix sort melhora as ordenações por comparação, como heapsort e o mergesort, que necessitam de $\Omega(n \cdot \log n)$ comparações, onde "n" é o número de itens a serem ordenados. Em compensação, algoritmos de ordenação baseados em comparações oferecem flexibilidade por serem aptos a ordenar de outras formas que não a lexicográfica. No entanto, essa habilidade é de pouca importância em várias aplicações práticas.

Radix Sort

Complexidade de Tempo: $\Theta(nk)$.

Complexidade de espaço: $\Theta(n + s)$.

- n = número de elementos.
- k = tamanho string.
- s = tamanho do alfabeto.

Equipe

- ▣ Marcos Oliveira
- ▣ Jose Antonio
- ▣ Maxwell Alves

Referências

https://pt.wikipedia.org/wiki/Radix_sort

https://en.wikipedia.org/wiki/Radix_sort