Radixsort

- Algumas vezes, a comparação entre chaves pode ser uma operação muito cara
- Existem alguns algoritmos de ordenação sem comparação de chaves
  - Bucket Sort
  - Counting Sort
  - Tally Sort
  - etc...
- **Radix Sort** é uma classe de algoritmos que usa a representação binária das chaves para a ordenação

Radixsort

- Idéia Geral:
  - chaves cujo bit mais a esquerda é 0, vem antes que chaves cujo bit é 1
  - Repetindo-se isso para todos os bits de forma adequada é possível ordenar
- Como extrair os bits
  - Formas eficientes em c: and (&) e shift(>>)
    - bit = x & 00000001; x >> 1;
  - Formas simples: divisão e resto
    - Considerando bits indexados de 0 a n da dir para esq, para extrair o bit i de um número X, temos **$(X / 2^i) \% 2$**

Radix Exchange Sort

- Algoritmo analisa os bits da esquerda para a direita
- Funcionamento similar ao do Quicksort, mas a partição é feita comparando-se bits ao invés de chaves
- Chamadas recursivas ordenando os subvetores pelo bit i-1
- Complexidade:
  - *O(n * b)* comparações de bits

Radix Exchange Sort

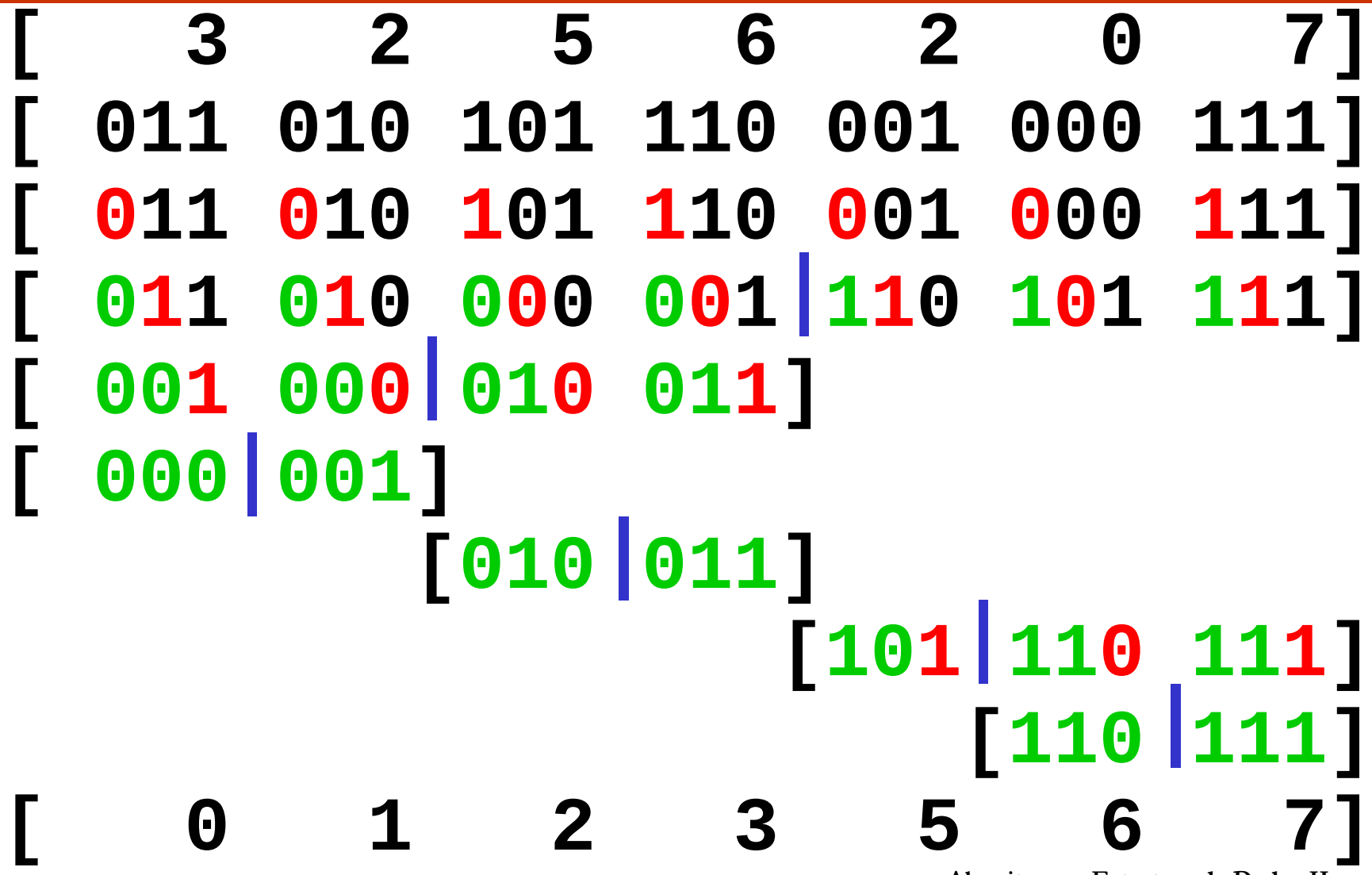
```
quicksortB(int a[], int l, int r, int w) {
  int i = l, j = r;

  if (r <= l || w > 0) return;
  while (j != i) {
      while (digit(a[i], w) == 0 && (i < j)) i++;
      while (digit(a[j], w) == 1 && (j > i)) j--;

      exch(a[i], a[j]);
  }
  if (digit(a[r], w) == 0) j++;
  quicksortB(a, l, j-1, w+1);
  quicksortB(a, j, r, w+1);
 }

void sort(Item a[], int l, int r) {
   quicksortB(a, l, r, numbits - 1);
}
```

Fonte: Algorithms in C
Robert Sedgewick

Exemplo

```
[   3   2   5   6   2   0   7]
[ 011 010 101 110 001 000 111]
```

Exemplo


[3 2 5 6 2 0 7]
[011 010 101 110 001 000 111]
[011 010 101 110 001 000 111]
[011 010 000 001|110 101 111]
[001 000|010 011]
[000|001]
[010|011]
[101|110 111]
[110|111]
[0 1 2 3 5 6 7]