
Comparação entre os Métodos de Ordenação Parcial

1. Para valores de k até 1.000, o método da InserçãoParcial é imbatível.
2. O QuicksortParcial nunca ficar muito longe da InserçãoParcial.
3. Na medida em que o k cresce, o QuicksortParcial é a melhor opção.
4. Para valores grandes de k , o método da InserçãoParcial se torna ruim.
5. Um método indicado para qualquer situação é o QuicksortParcial.
6. O HeapsortParcial tem comportamento parecido com o do QuicksortParcial.
7. No entanto, o HeapsortParcial é mais lento.

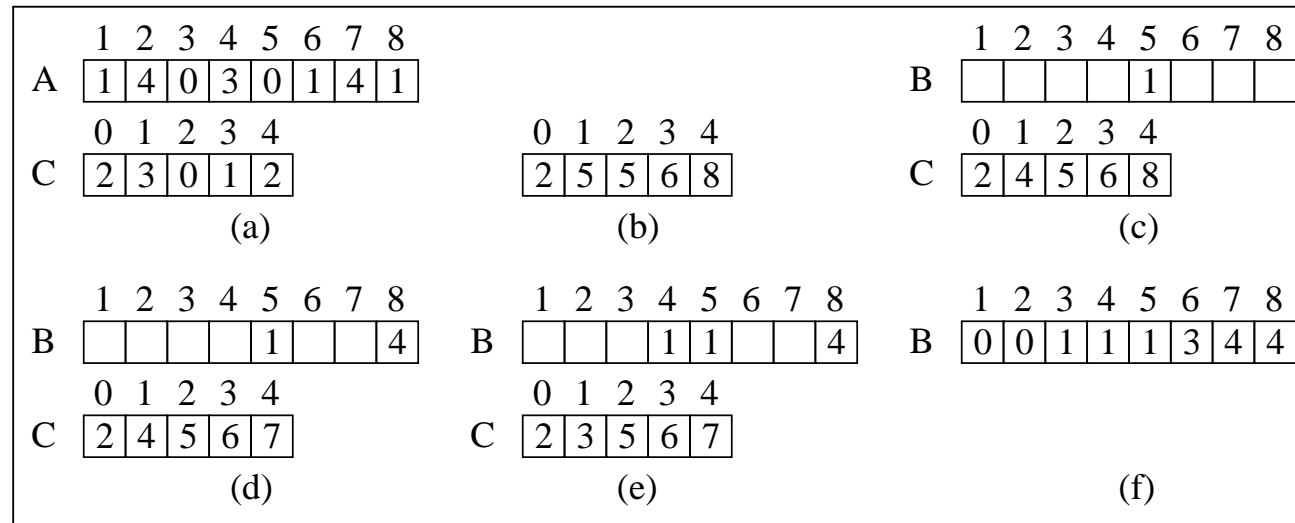
Ordenação em Tempo Linear

- Nos algoritmos apresentados a seguir não existe comparação entre chaves.
- Eles têm complexidade de tempo linear na prática.
- Necessitam manter uma cópia em memória dos itens a serem ordenados e uma área temporária de trabalho.

Ordenação por Contagem

- Este método assume que cada item do vetor A é um número inteiro entre 0 e k .
- O algoritmo conta, para cada item x , o número de itens antes de x .
- A seguir, cada item é colocado no vetor de saída na sua posição definitiva.

Ordenação por Contagem



- A contém oito chaves de inteiros entre 0 e 4. Cada etapa mostra:
 - (a) o vetor de entrada A e o vetor auxiliar C contendo o número de itens iguais a i , $0 \leq i \leq 4$;
 - (b) o vetor C contendo o número de itens $\leq i$, $0 \leq i \leq 4$;
 - (c), (d), (e) os vetores auxiliares B e C após uma, duas e três iterações, considerando os itens em A da direita para a esquerda;
 - (f) o vetor auxiliar B ordenado.

Ordenação por Contagem

```
void Contagem(TipoItem *A, TipoIndice n, int k)
{ int i;
  for (i = 0; i <= k; i++) C[i] = 0;
  for (i = 1; i <= n; i++) C[A[i].Chave] = C[A[i].Chave] + 1;
  for (i = 1; i <= k; i++) C[i] = C[i] + C[i-1];
  for (i = n; i > 0; i--)
  { B[C[A[i].Chave]] = A[i];
    C[A[i].Chave] = C[A[i].Chave] - 1;
  }
  for (i = 1; i <= n; i++)
    A[i] = B[i];
}
```

Ordenação por Contagem

- Os arranjos auxiliares B e C devem ser declarados fora do procedimento Contagem para evitar que sejam criados a cada chamada do procedimento.
- No quarto for, como podem haver itens iguais no vetor A , então o valor de $C[A[j]]$ é decrementado de 1 toda vez que um item $A[j]$ é colocado no vetor B . Isso garante que o próximo item com valor igual a $A[j]$, se existir, vai ser colocado na posição imediatamente antes de $A[j]$ no vetor B .
- O último for copia para A o vetor B ordenado. Essa cópia pode ser evitada colocando o vetor B como parâmetro de retorno no procedimento Contagem, como mostrado no Exercício 4.24.
- A ordenação por contagem é um método estável.