unipac.br
Barbacena

Bacharelado em Ciência da Computação

Estruturas de Dados

Material de Apoio

Parte XIV – *Método CountingSort*

Prof. Nairon Neri Silva
naironsilva@unipac.br

2º sem / 2021

Algoritmo Countingsort

- A ordenação por contagem pressupõe que cada um dos ***n*** elementos do vetor de entrada é um inteiro entre 1 e ***k*** (***k*** representa o maior inteiro presente no vetor).

- A ideia básica é determinar, para cada elemento de entrada ***x***, o número de elementos menores ou iguais a ***x***. Assim é possível determinar exatamente onde o elemento ***x*** será inserido.

Algoritmo Countingsort

- Por exemplo, se 17 elementos são menores do que x, então na posição x temos o valor 18 da saída.

- O método pode ser dividido em 4 passos:
  - Inicialização dos vetores
  - Contagem do número de ocorrências de cada elemento
  - Contagem da quantidade de números menores
  - Ordenação

Algoritmo Countingsort

- O Countingsort faz uso de 3 vetores:
  - vetA: um vetor de tamanho n que será o vetor a ser ordenado;
  - vetB: um vetor de tamanho n que será o vetor utilizado para a ordenação dos elementos;
  - vetC: um vetor de tamanho ***k*** (valor do maior elemento do vetor de entrada), que será utilizado para a contagem da quantidade de vezes que cada elemento aparece no vetor de entrada e a posição onde ele deve ser inserido no vetor ordenado.

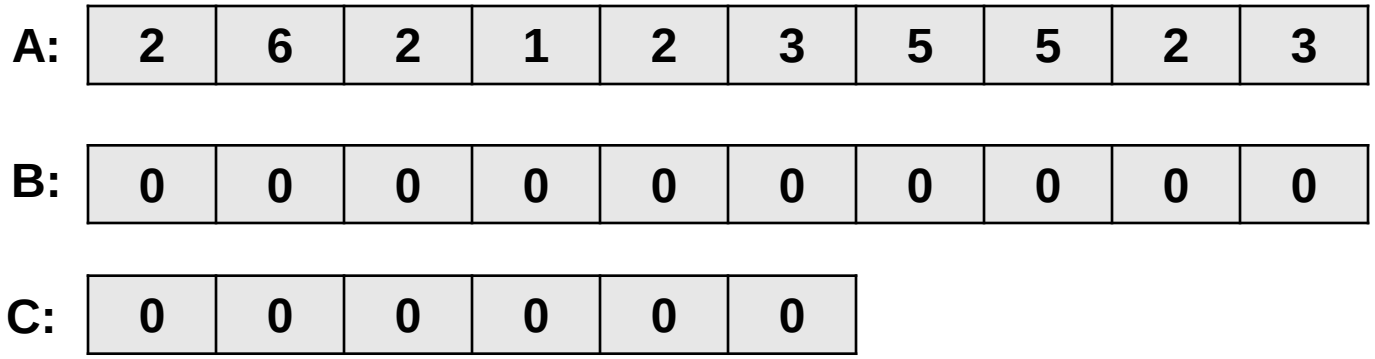
Algoritmo Countingsort

```
void countingSort(int v[], int n) {
    int i, ordenado[n];

    // encontra o maior elemento do vetor
    int maior = v[0];
    for (i = 1; i < n; i++) {
        if (v[i] > maior)
            maior = v[i];
    }

    // cria o vetor que faz a soma das ocorrências de cada número
    int contagem[maior];

    // inicializa o vetor com zeros
    for (i = 0; i <= maior; ++i) {
        contagem[i] = 0;
    }
```

Algoritmo Countingsort

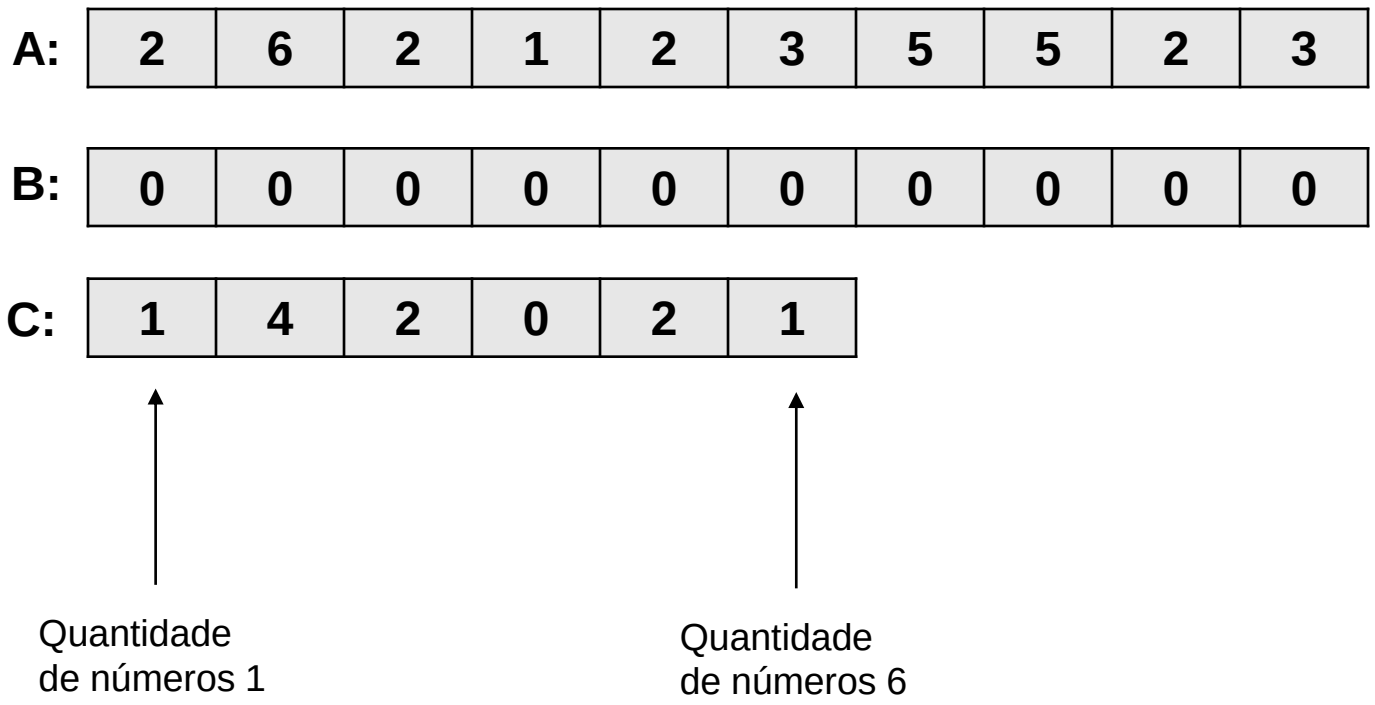
```
    // guarda a contagem de cada elemento
    for (i = 0; i < n; i++) {
        contagem[v[i]]++;
    }

    // guarda a contagem cumulativa
    for (i = 1; i <= maior; i++) {
        contagem[i] += contagem[i - 1];
    }

    // ordena os dados de acordo com os índices presentes no vetor
    for (int i = n - 1; i >= 0; i--) {
        ordenado[contagem[v[i]] - 1] = v[i];
        contagem[v[i]]--;
    }

    // copia os dados do vetor ordenado para o vetor original
       //passado como parâmetro
    for (int i = 0; i < n; i++) {
        v[i] = ordenado[i];
    }
}
```

Algoritmo Countingsort - Inicialização

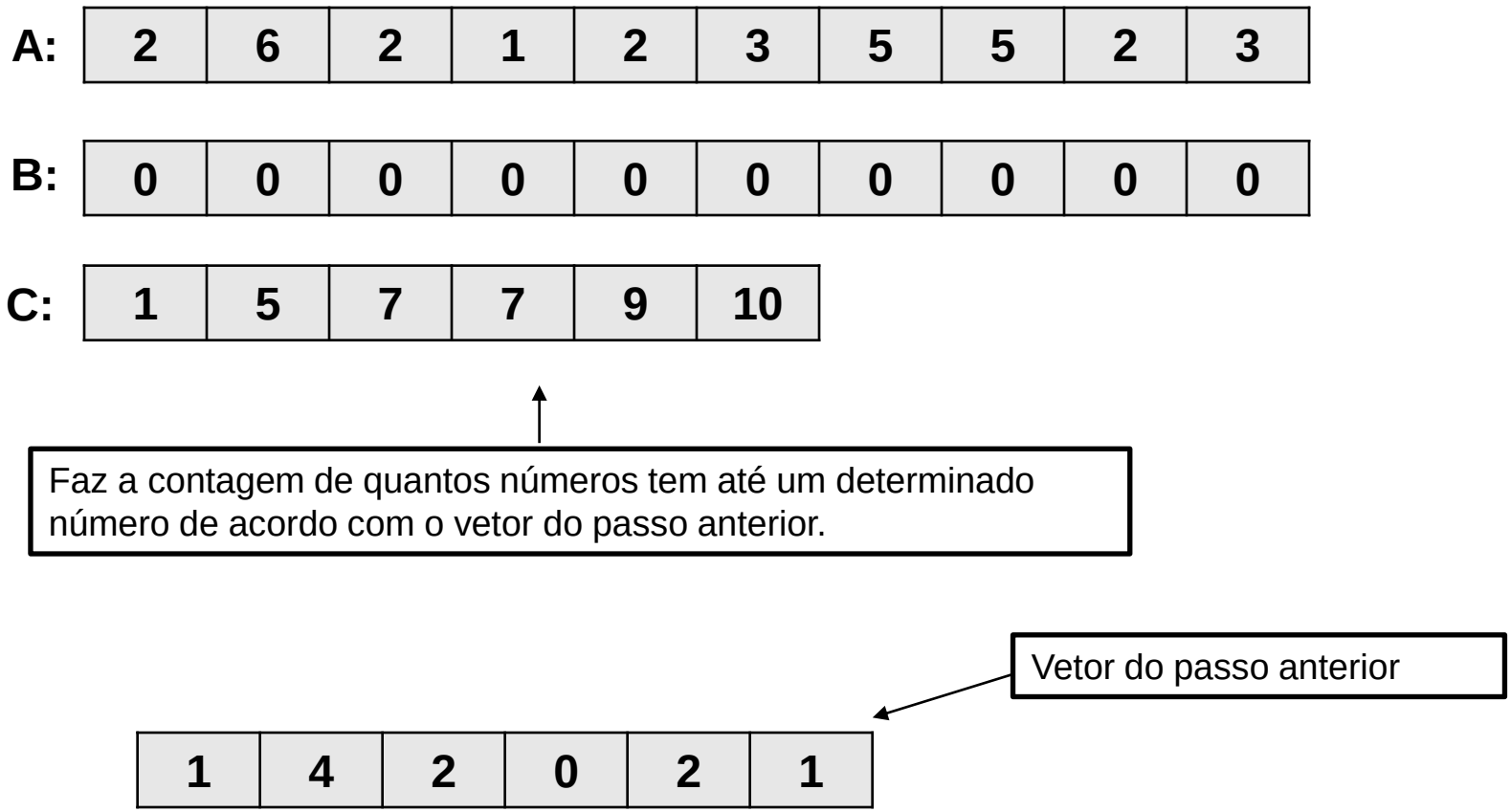


- A: vetor a ser ordenado (tamanho ***10***)
- B: vetor para a ordenação (tamanho ***10***)
- C: vetor para contagem (menor valor é o 1 e o maior é o 6, logo, precisa-se de um vetor de 6 posições)

Algoritmo Countingsort – Contagem de Ocorrências



Algoritmo Countingsort – Contagem de nº menores



Algoritmo Countingsort – Ordenação

A:	**2**	**6**	**2**	**1**	**2**	**3**	**5**	**5**	**2**	**3**
B:	**1**	**2**	**2**	**2**	**2**	**3**	**3**	**5**	**5**	**6**
C:	**0**	**1**	**5**	**7**	**7**	**9**				

A ordenação consiste em verificar a posição de um determinado número, coloca-lo na posição indicada no vetor C e subtrair -1 da posição

Algoritmo Countingsort – Análise

- É um algoritmo estável?
- Desvantagens (pelo menos três).
- É vantajoso em relação ao algoritmos que realizam trocas?

Algoritmo Countingsort – Análise

- Vantagens:
  - Algoritmo eficiente se ***k*** não for muito grande.
  - Não realiza comparações;
  - É um algoritmo de ordenação estável;

- Desvantagens:
  - Necessita de dois vetores adicionais para sua execução, utilizando, assim, mais espaço na memória.
  - Não indicado para chaves de caracteres.
  - Se ***k*** for um valor muito grande não será eficiente.

Referências

- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Makron books.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto Castilho. *Algoritmos e estruturas de dados*. LTC Editora.
- FIDALGO, Robson. *Material para aulas*. UFRPE.
- NELSON, Fábio. *Material para aulas: Algoritmo e Programação.* UNIVASP.
- FEOFILOFF, P., *Algoritmos em linguagem C*, Editora Campus, 2008.
- ZIVIANI, N., *Projeto de algoritmos com Implementações em Pascal e C*, São Paulo: Pioneira, 2d, 2004.
- http://www.ime.usp.br/~pf/algoritmos/
- MELLO, Ronaldo S., *Material para aulas: Ordenação de Dados*, UFSC-CTC-INE
- MENOTTI, David, *Material para aulas: Algoritmos e Estrutura de Dados I*, DECOM-UFOP