Counting Sort

SCC0201 - Introdução à Ciência de Computação II

Clausius G. Reis Leandro A. Amaral Tiago S. Nazaré
Vanessa Q. Marinho

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

21 de junho de 2015

Agenda

Agenda

Ideia Básica

Ordena objetos de acordo com as chaves **inteiras**.

- É um algoritmo de ordenação inteira.

Conta o número de objetos que têm uma mesma chave.

Deve haver um número limitado de possíveis chaves.

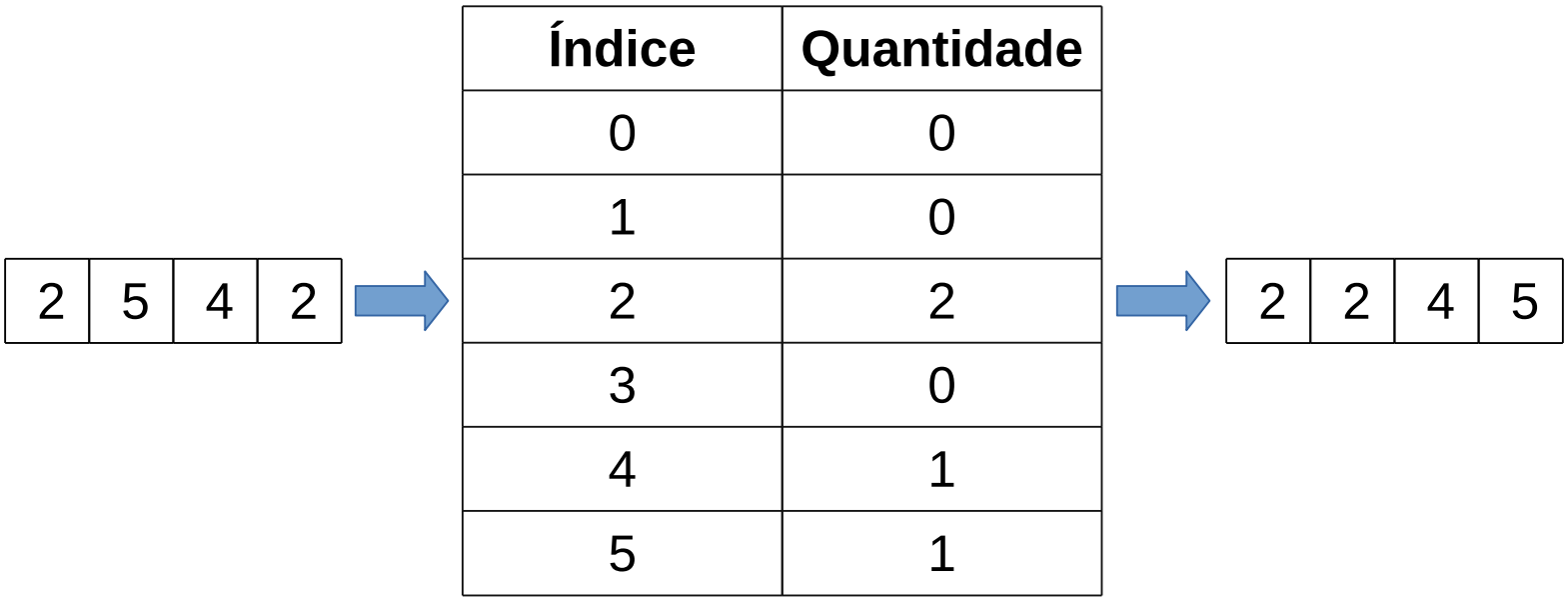
- Para que seu tempo de execução seja baixo, o tamanho do intervalo dos valores das chaves não pode ser muito maior que o número de elementos a serem ordenados.

Agenda

Primeira Versão - Simplificada

Computa a quantidade de cada elemento e gera um vetor de saída que tenha as quantidades necessárias de cada elemento.

Exemplo: Ordenar um vetor só pode ter valores entre 0 e 5



Primeira Versão - Código

```
void simple_counting_sort(int *v, int n, int k){
        int *count = (int *)malloc(k*sizeof(int));
        int i, j;

        for(i = 0; i < k; ++i){
                count[i] = 0;
        }

        for(i = 0; i < n; ++i){
                ++count[v[i]];
        }

        for(i = 0; i < k; ++i){
                for(j = 0; j < count[i]; ++j){
                        printf("%d ", i);
                }
        }
        printf("\n");

        free(count);
}
```

Imagine que tem-se um grupo de alunos e suas respectivas quantidades de faltas nas últimas 3 aulas:

$$[\{0, Ana\}, \{1, José\}, \{1, João\}, \{0, Maria\}]$$

Questão: O algoritmo anterior conseguiria ordenar os alunos por número de faltas? Por quê?

Agenda

Segunda Versão

O algoritmo anterior considera somente as chaves e não os objetos

- Não consegue ordenar os alunos por número de faltas.

Estratégia:

1. Contar o número de chaves (faltas) do mesmo valor;
2. Com base nos contadores calcular a menor posição do vetor ordenado que cada valor de chave irá ocupar;
3. Percorrer o vetor original e, com base no vetor de posições, montar o vetor ordenado.

Segunda Versão - Exemplo

$$[\{0, Ana\}, \{1, José\}, \{1, João\}, \{0, Maria\}]$$

Passo 1: Contar o número de chaves (faltas) do mesmo valor;

Faltas	**Alunos**
0	2
1	2
2	0
3	0

Segunda Versão - Exemplo

Faltas	Alunos
0	2
1	2
2	0
3	0

Passo 2: Menor posição do vetor ordenado para cada chave

- Pode ser calculada somando-se o total de itens com chave menor que a atual

Faltas (F)	Índice	Cálculo
0	0	-
1	2	$F(0) = 2$
2	4	$F(0) + F(1) = 2 + 2 = 4$
3	4	$F(0) + F(1) + F(3) = 2 + 2 + 0 = 4$

Segunda Versão - Exemplo

Passo 3: Montar o vetor ordenado

$$[\{\mathbf{0}, \mathbf{Ana}\}, \{1, \textit{José}\}, \{1, \textit{João}\}, \{0, \textit{Maria}\}]$$

Faltas	**Índice**
0	0
1	2
2	4
3	4

$$[-, -, -, -]$$

Segunda Versão - Exemplo

Passo 3: Montar o vetor ordenado

$$[\{0, Ana\}, \{\mathbf{1}, \mathbf{José}\}, \{1, João\}, \{0, Maria\}]$$

Faltas	**Índice**
0	$0 + 1 = 1$
1	2
2	4
3	4

$$[\{0, Ana\}, -, -, -]$$

Segunda Versão - Exemplo

Passo 3: Montar o vetor ordenado

$$[\{0, Ana\}, \{1, José\}, \{\mathbf{1}, \mathbf{João}\}, \{0, Maria\}]$$

Faltas	**Índice**
0	$0 + 1 = 1$
1	$2 + 1 = 3$
2	4
3	4

$$[\{0, Ana\}, -, \{1, José\}, -]$$

Segunda Versão - Exemplo

Passo 3: Montar o vetor ordenado

$$[\{0, Ana\}, \{1, José\}, \{1, João\}, \{\mathbf{0}, \mathbf{Maria}\}]$$

Faltas	**Índice**
0	$0 + 1 = 1$
1	$2 + 1 + 1 = 4$
2	4
3	4

$$[\{0, Ana\}, -, \{1, José\}, \{1, João\}]$$

Segunda Versão - Exemplo

Passo 3: Montar o vetor ordenado

$$[\{0, Ana\}, \{1, José\}, \{1, João\}, \{0, Maria\}]$$

Faltas	**Índice**
0	$0 + 1 + 1 = 2$
1	$2 + 1 + 1 = 4$
2	4
3	4

$$[\{\mathbf{0}, \mathbf{Ana}\}, \{\mathbf{0}, \mathbf{Maria}\}, \{\mathbf{1}, \mathbf{José}\}, \{\mathbf{1}, \mathbf{João}\}]$$

Análise de Complexidade

Para ordenar um vetor com n elementos e k possíveis chaves:

Primeira versão: $\mathcal{O}(n+k)$

- Inicializar o vetor de contagem (tamanho k) com zeros: $\mathcal{O}(k)$;
- Contar as ocorrências (percorrer o vetor desordenado): $\mathcal{O}(n)$;
- Gerar o vetor ordenado: $\mathcal{O}(n+k)$;

Segunda versão: $\mathcal{O}(n+k)$

- Inicializar o vetor de contagem (tamanho k) com zeros: $\mathcal{O}(k)$;
- Contar as ocorrências (percorrer o vetor desordenado): $\mathcal{O}(n)$;
- Calcular o primeiro índice de cada chave: $\mathcal{O}(k)$;
- Gerar o vetor ordenado: $\mathcal{O}(n)$;

Tarefa - Para a Próxima Aula

Implementar (em C) a segunda versão do algoritmo usando listas.

Assistir o seguinte vídeo:

- `https://www.youtube.com/watch?v=Nz1KZXbghj8`