

Counting Sort (0)

Implementierung



Zu sortierendes
Array

A

2	6	3	8	1	4	7	3	3	6
---	---	---	---	---	---	---	---	---	---

Zählerarray

C

[illegible]

Sortiertes Array

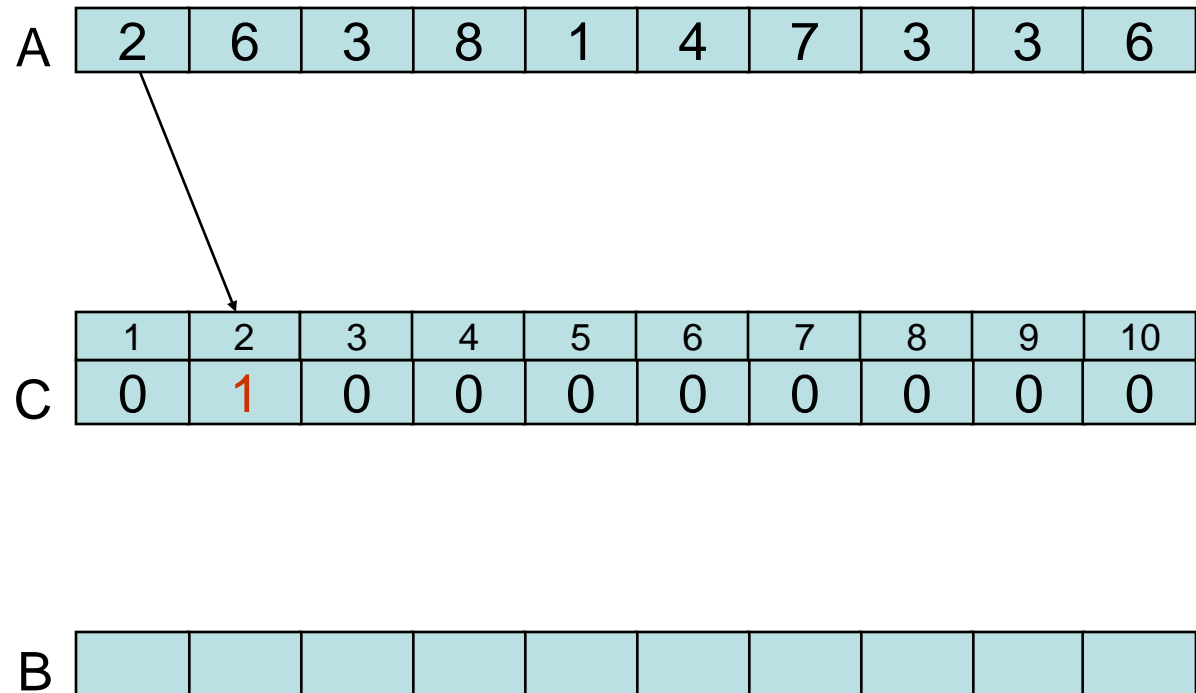
B

[illegible]

B

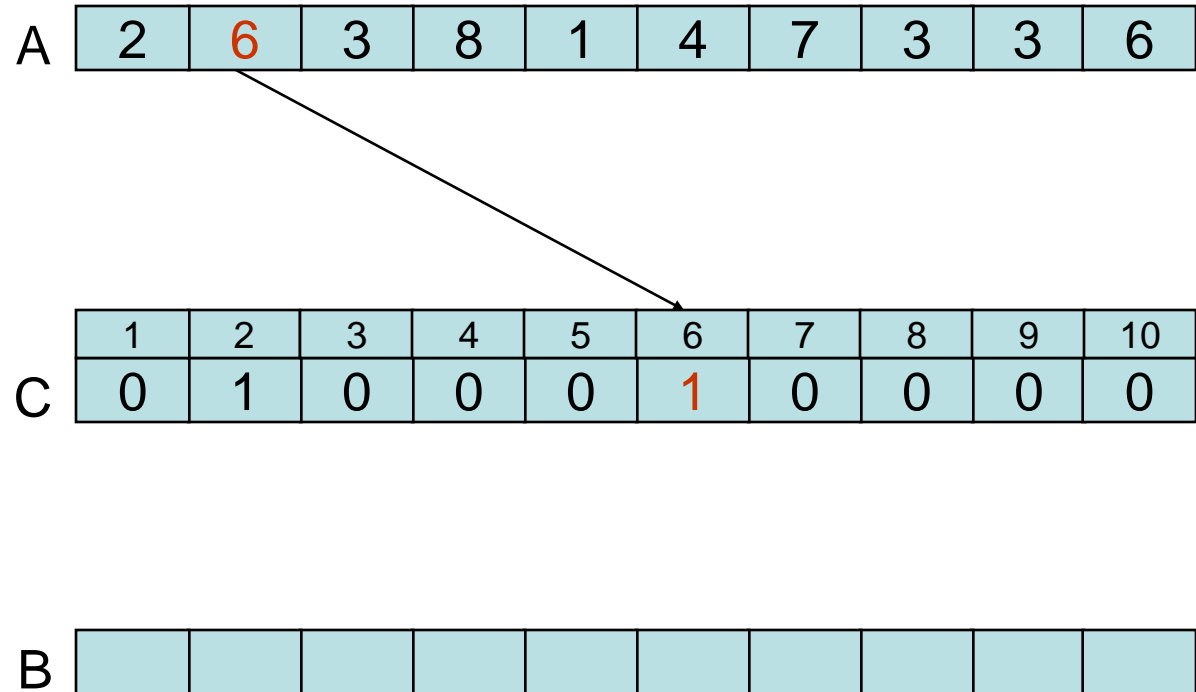
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



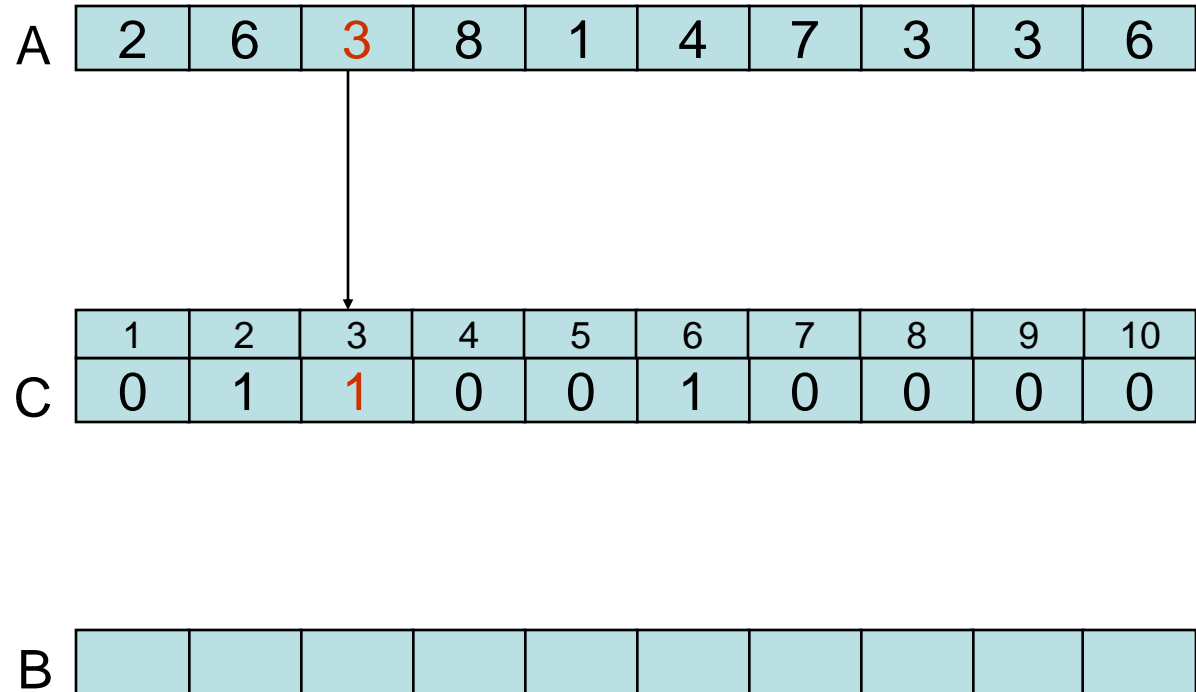
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



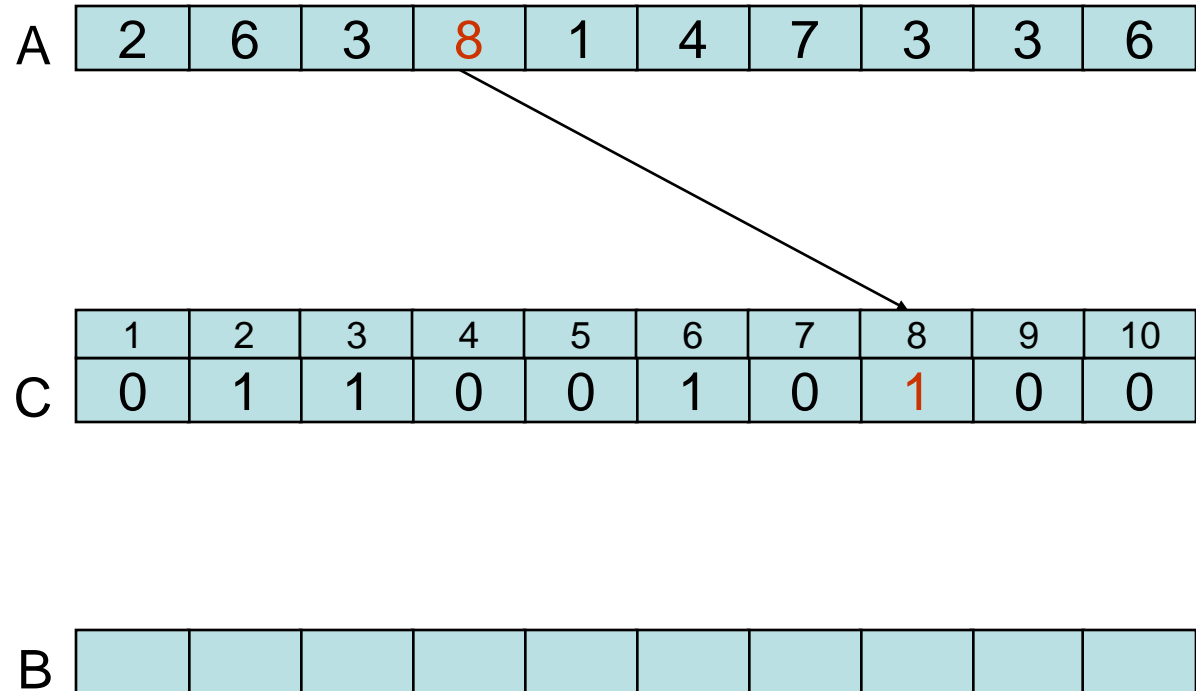
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



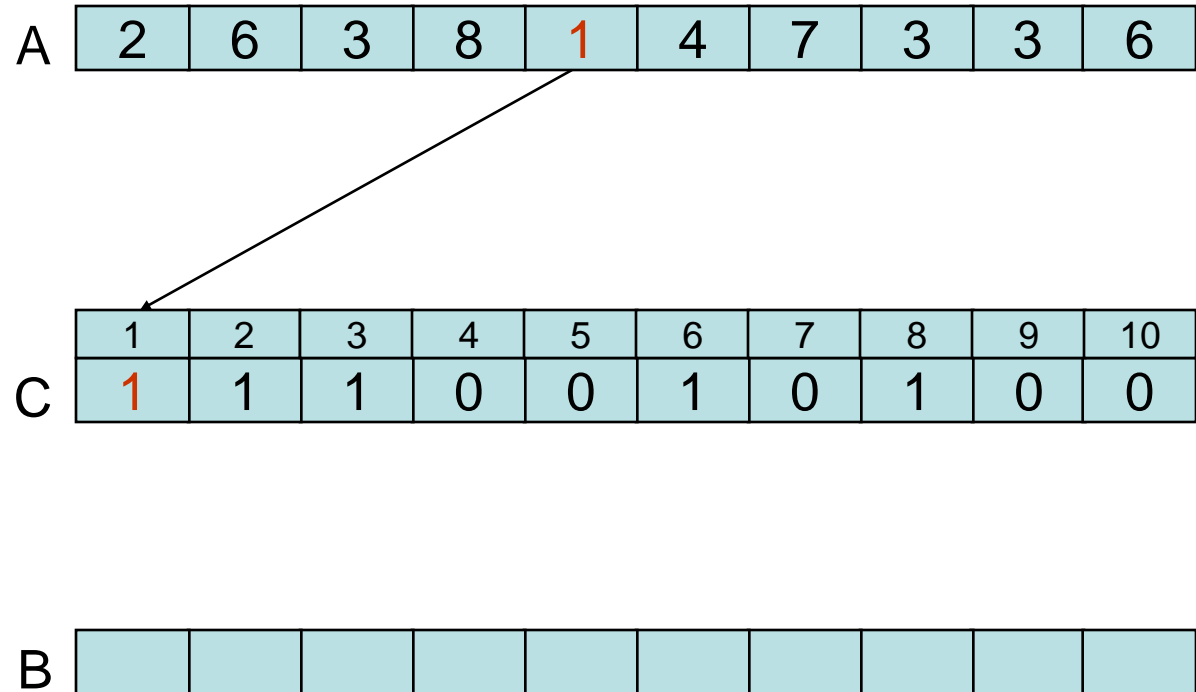
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



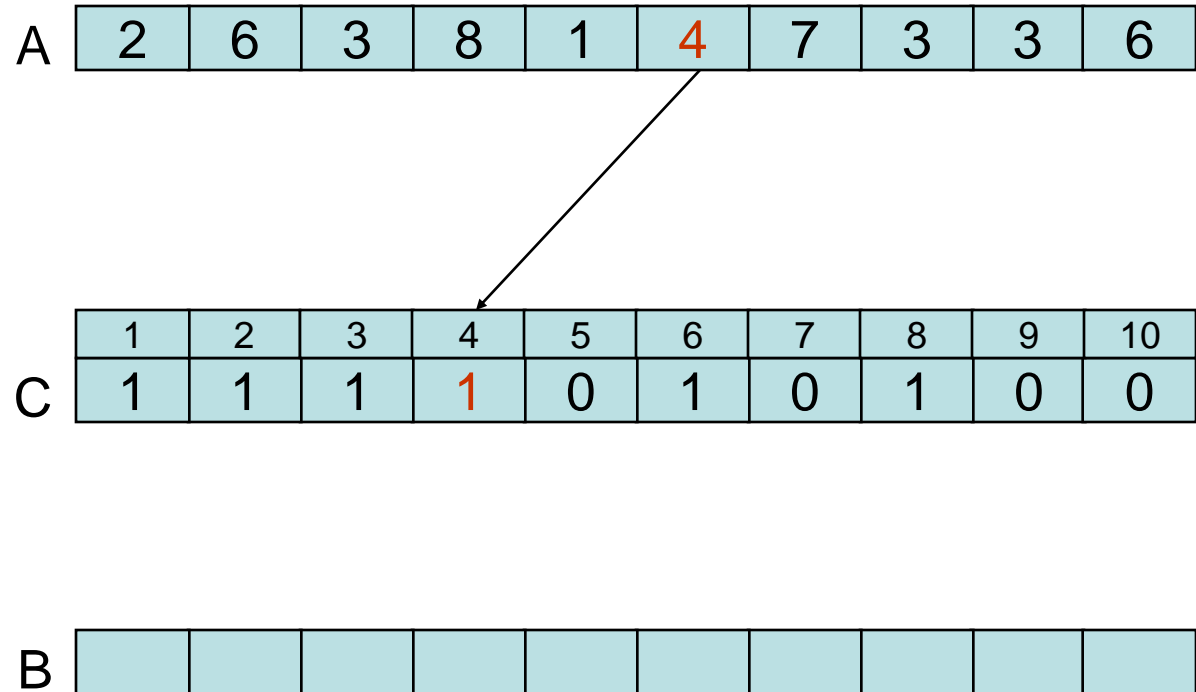
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



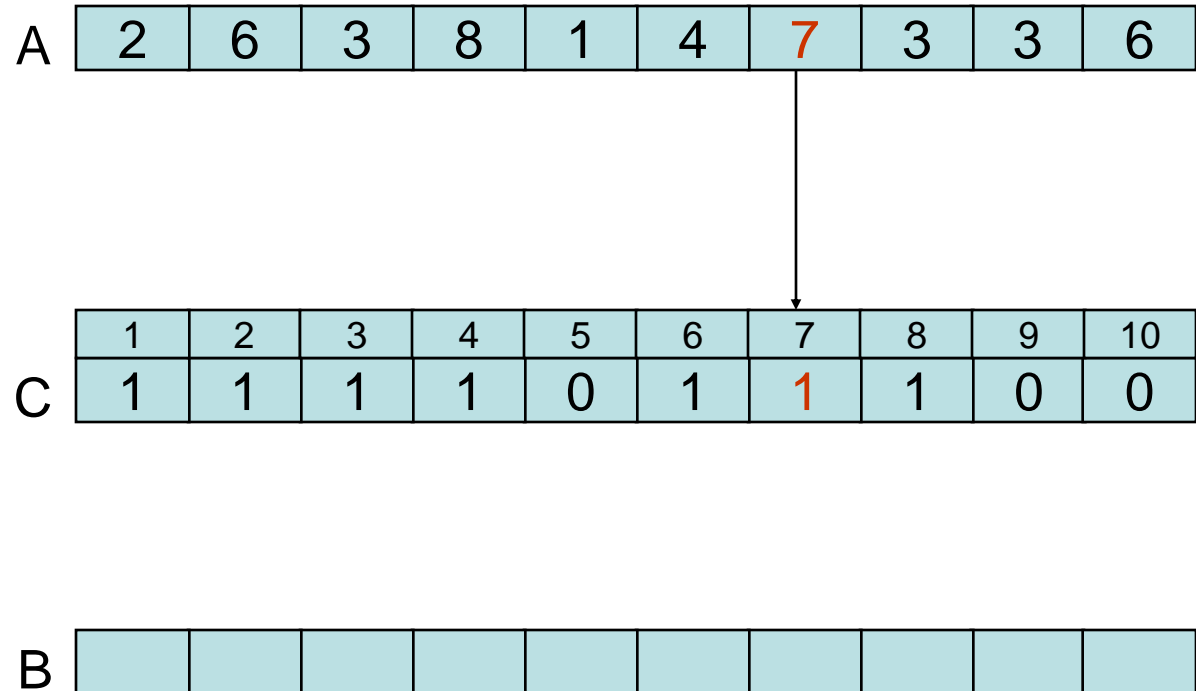
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



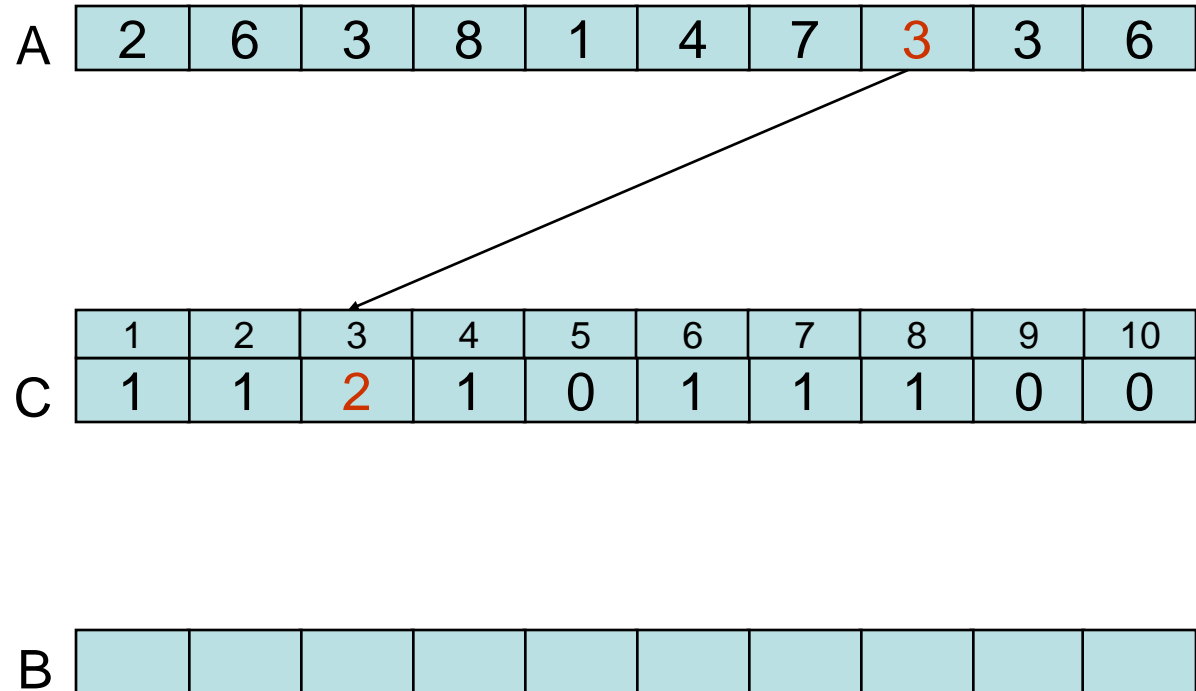
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



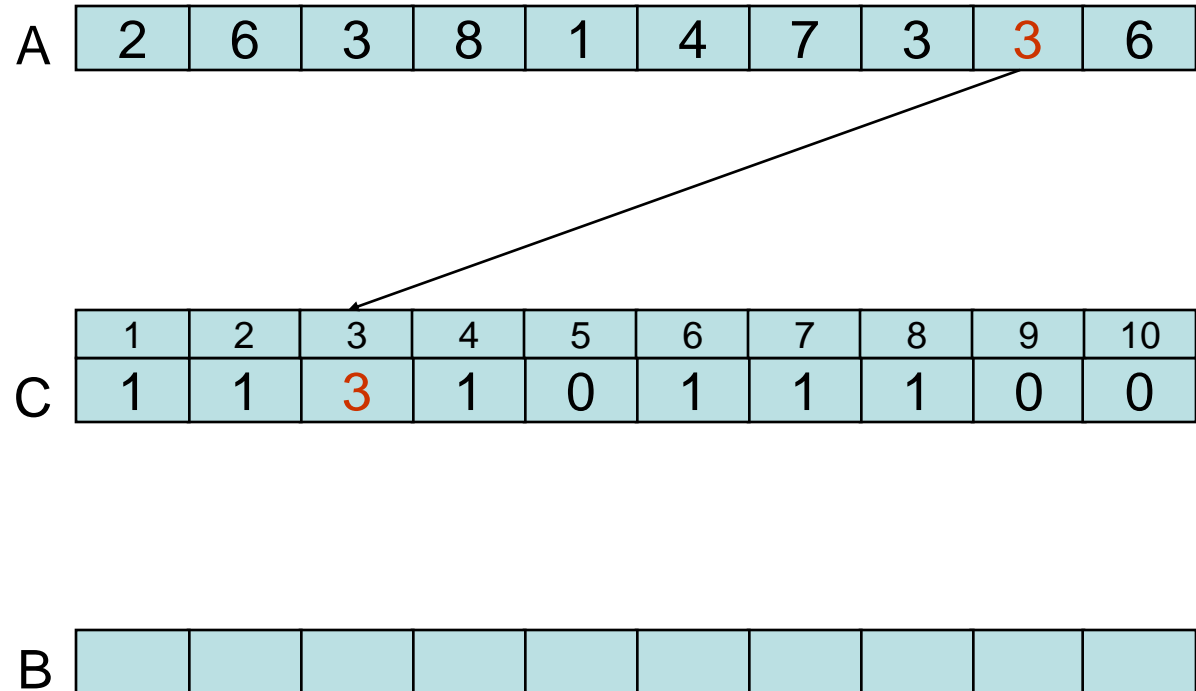
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



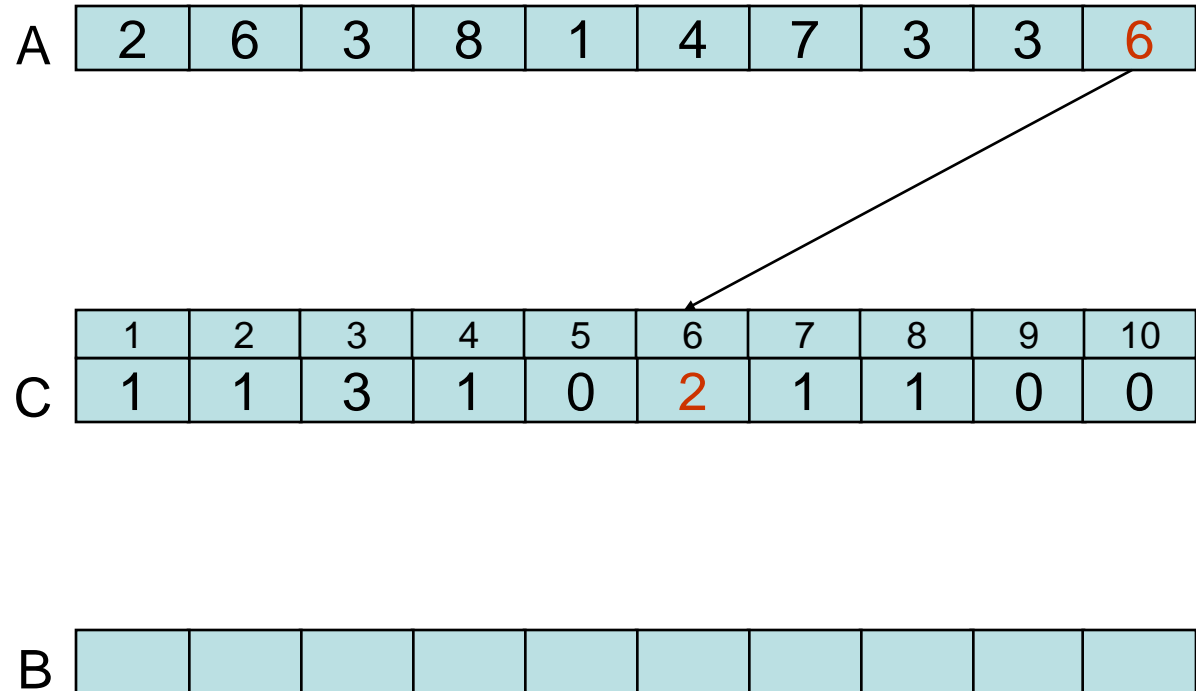
1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



1. Zählen der Vorkommnisse

```
for i = 1 to k  
  do C[i] = 0  
for j = 1 to length(A)  
  do C[A[j]] = C[A[j]] + 1
```



Counting Sort (16)

Implementierung



1. Zählen der Vorkommnisse
2. **Aufsummieren in B (Anzahl der Elemente die kleiner sind als $A[i]$)**

```

for i = 1 to k
    do C[i] = 0
for j = 1 to length(A)
    do C[A[j]]=C[A[j]] + 1
for i = 2 to k
    do C[i]=C[i]+C[i-1]

```

A	2	6	3	8	1	4	7	3	3	6
---	---	---	---	---	---	---	---	---	---	---

	1	2	3	4	5	6	7	8	9	10
C	1	2	5	6	6	2	1	1	0	0

B

B

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```

for i = 1 to k
    do C[i] = 0
for j = 1 to length(A)
    do C[A[j]]=C[A[j]] + 1
for i = 2 to k
    do C[i]=C[i]+C[i-1]
for j = length[A] downto 1 do
    B[C[A[j]]]= A[j]
    C[A[j]]=C[A[j]]-1

```

A	2	6	3	8	1	4	7	3	3	6
---	---	---	---	---	---	---	---	---	---	---

	1	2	3	4	5	6	7	8	9	10
C	1	2	5	6	6	8	9	10	10	10

B

Counting Sort (23)

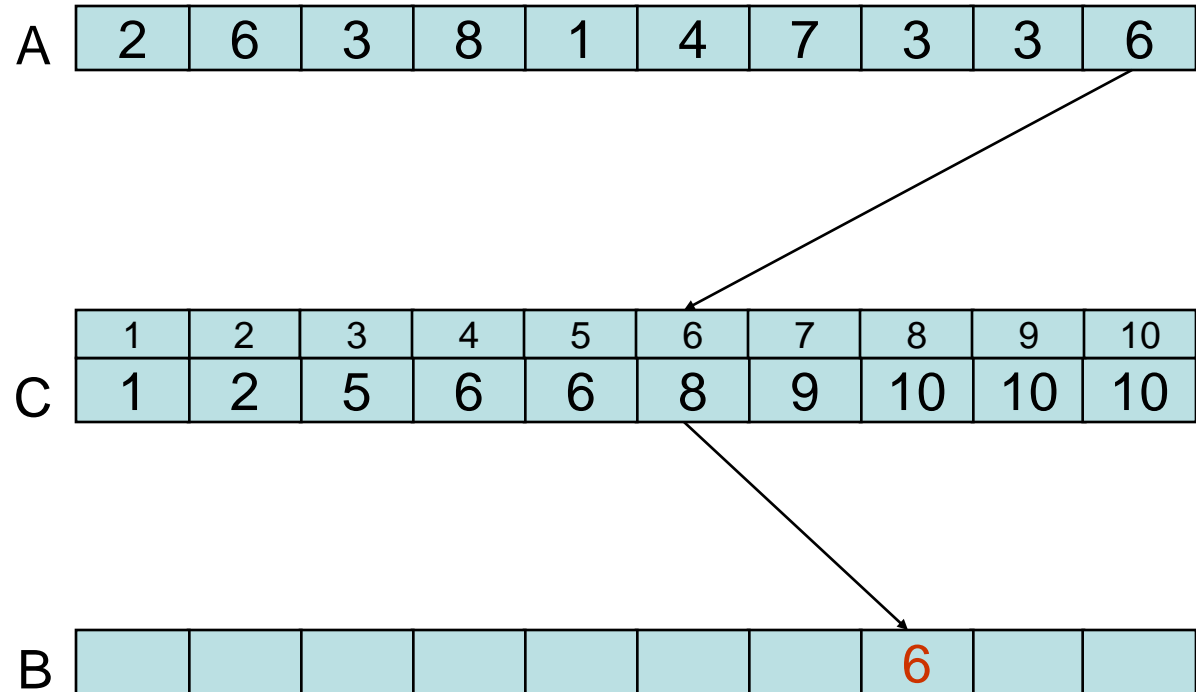
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (24)

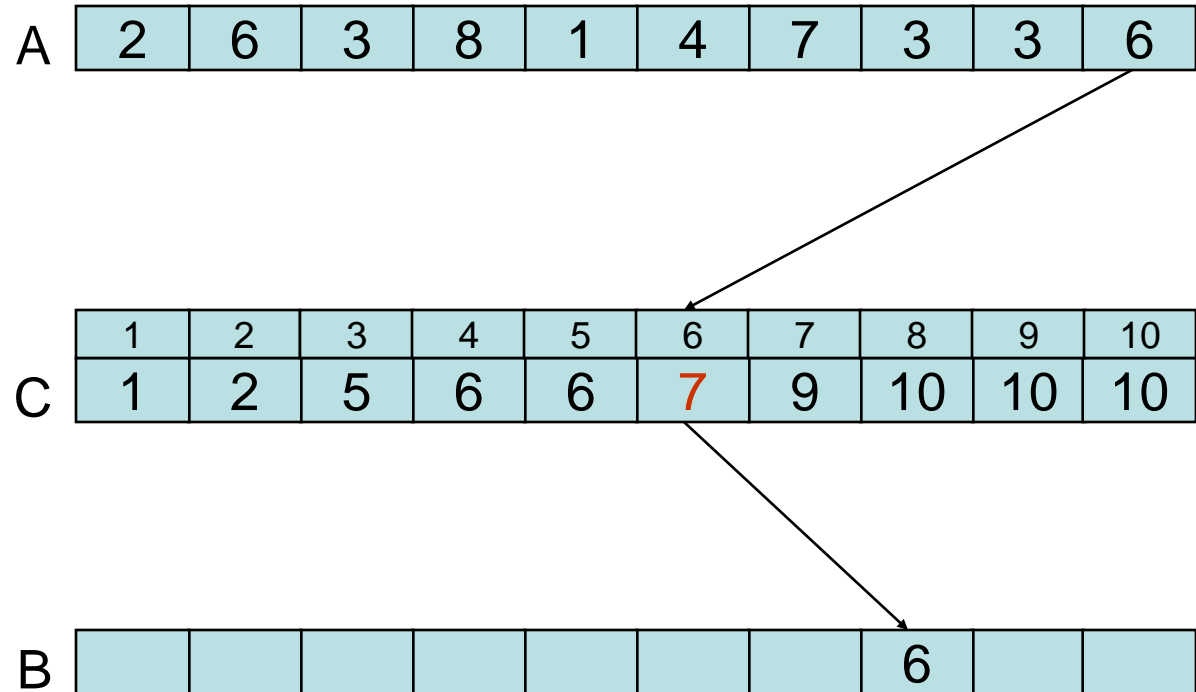
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (25)

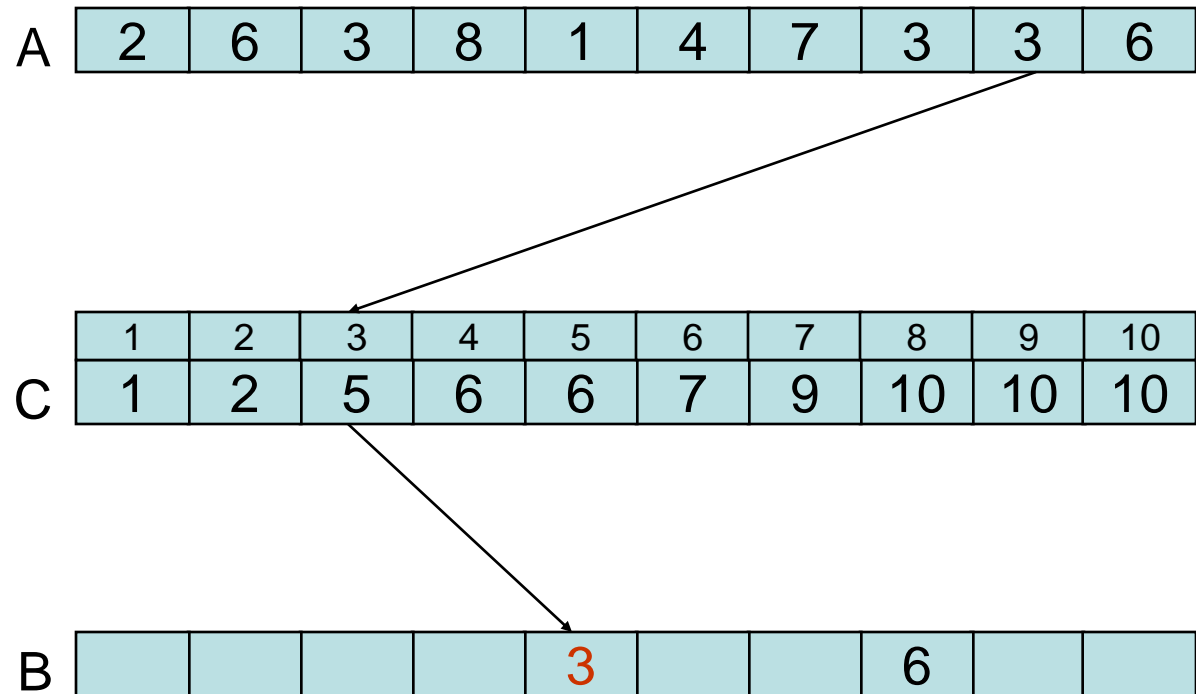
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (26)

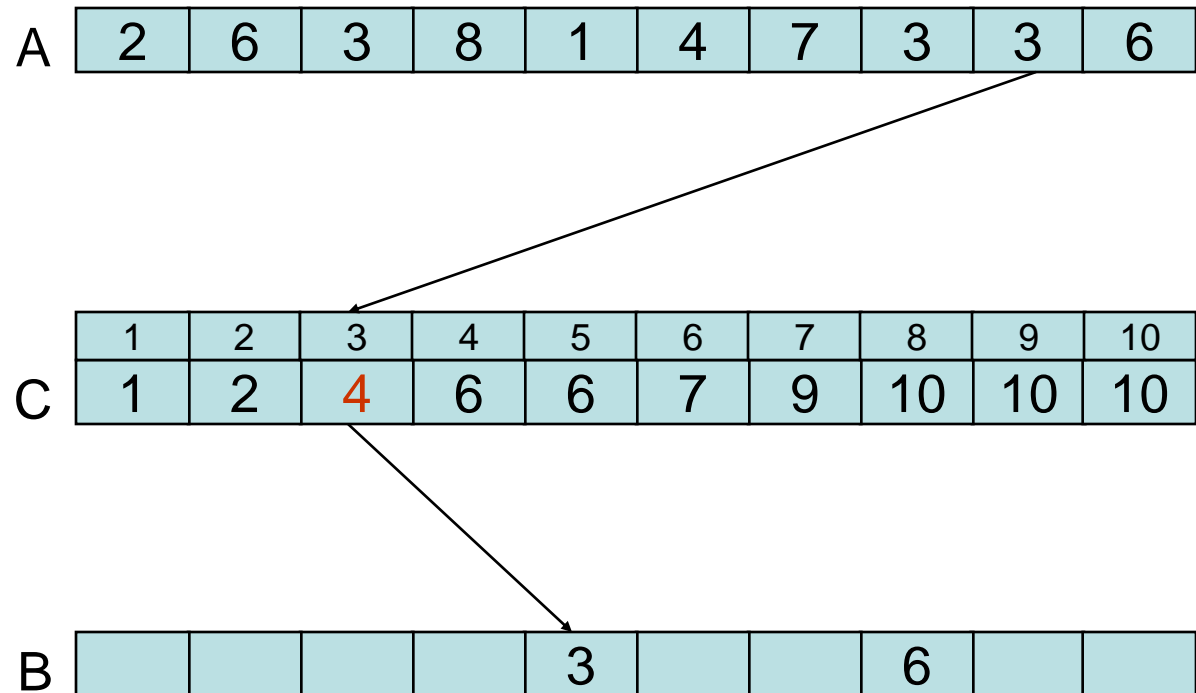
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (27)

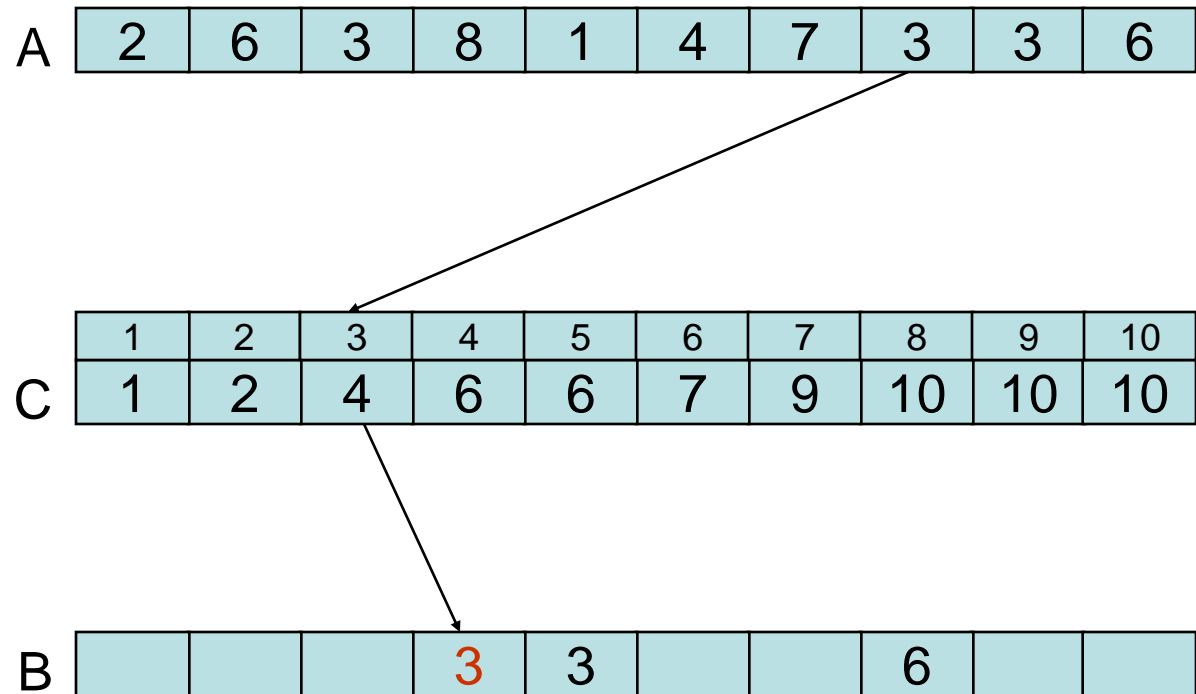
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (28)

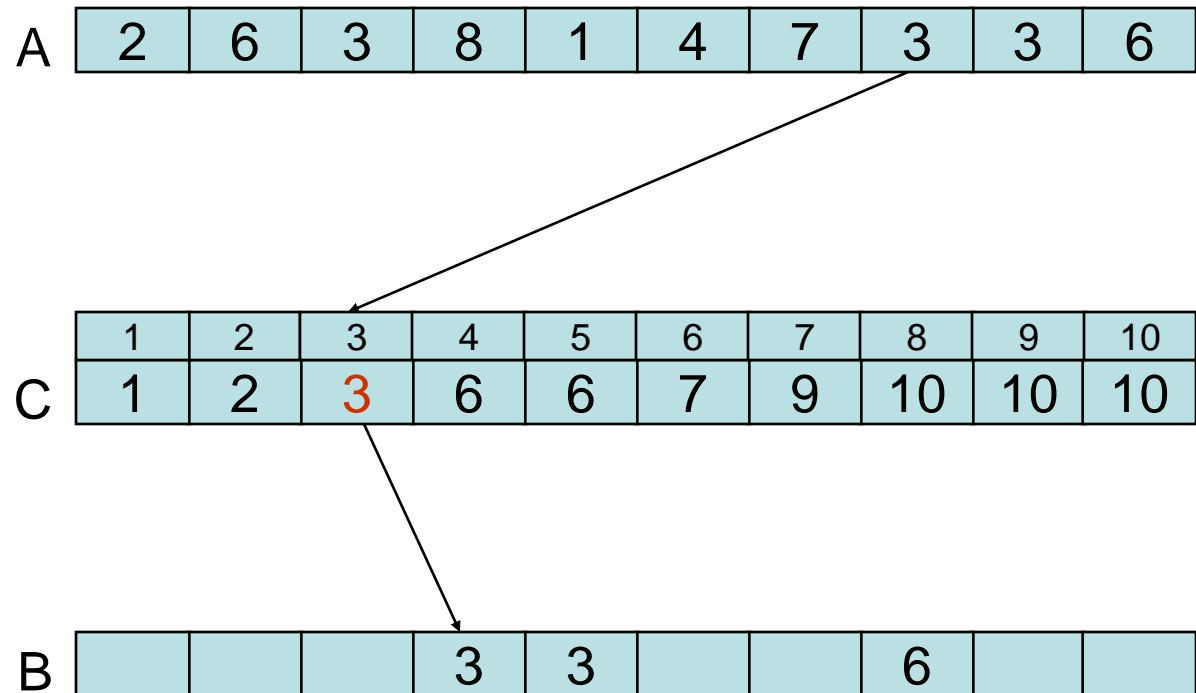
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (29)

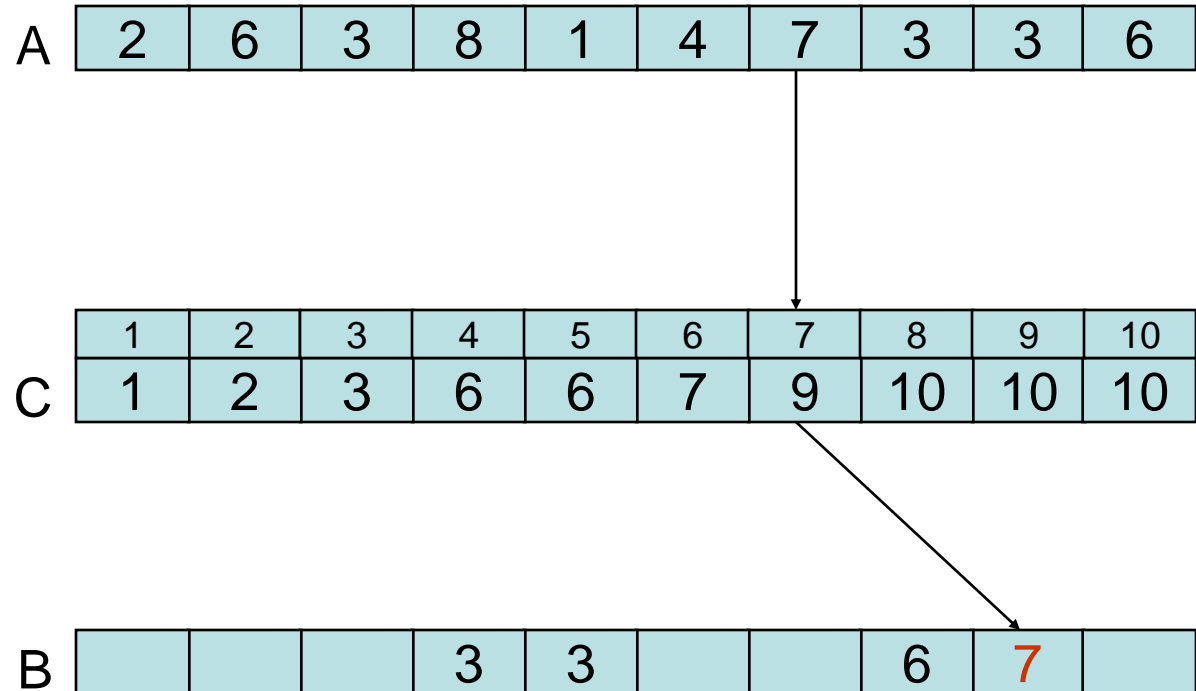
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (30)

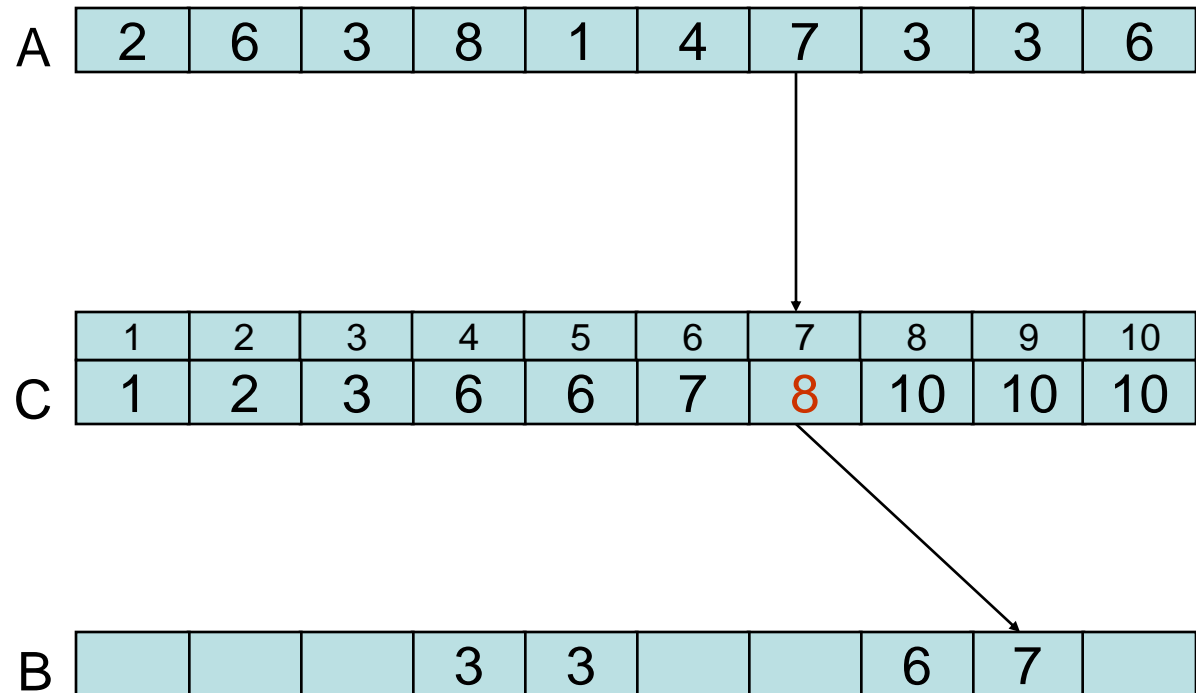
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (31)

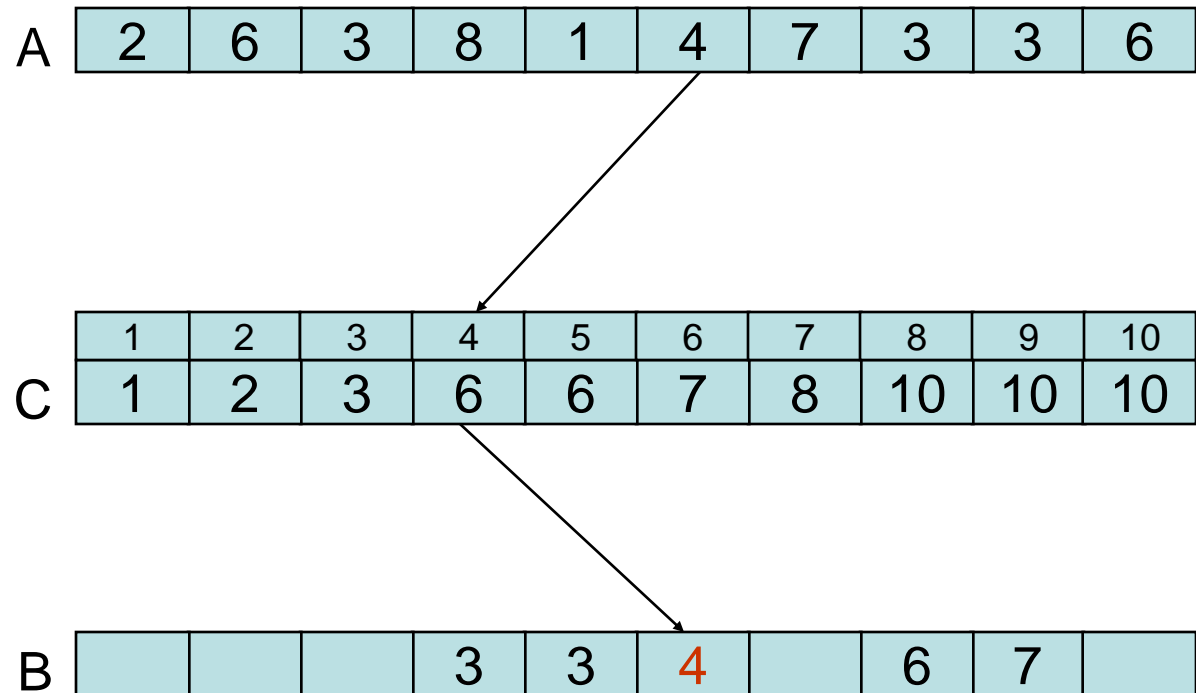
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (32)

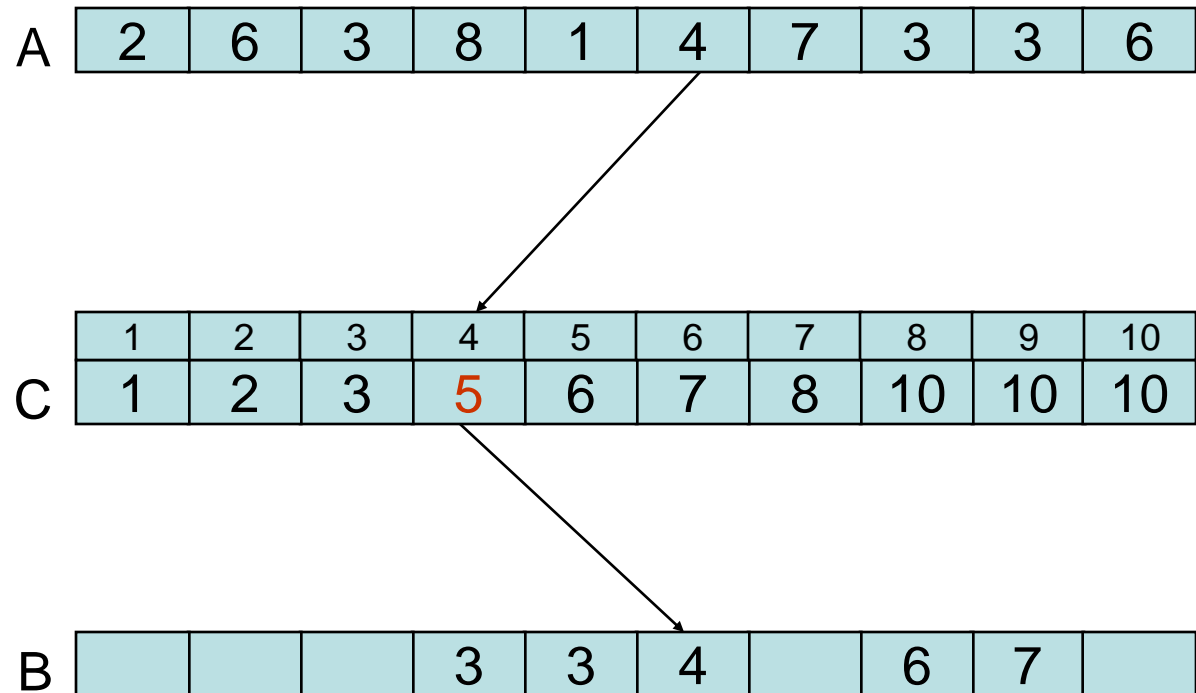
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (33)

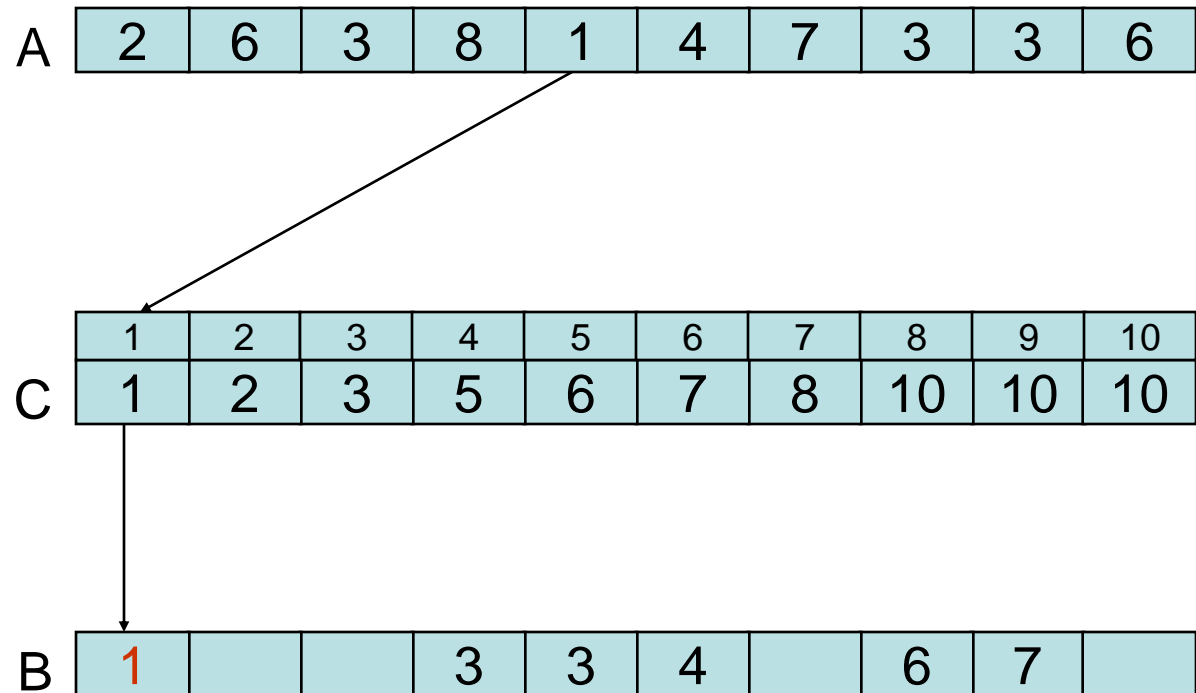
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (34)

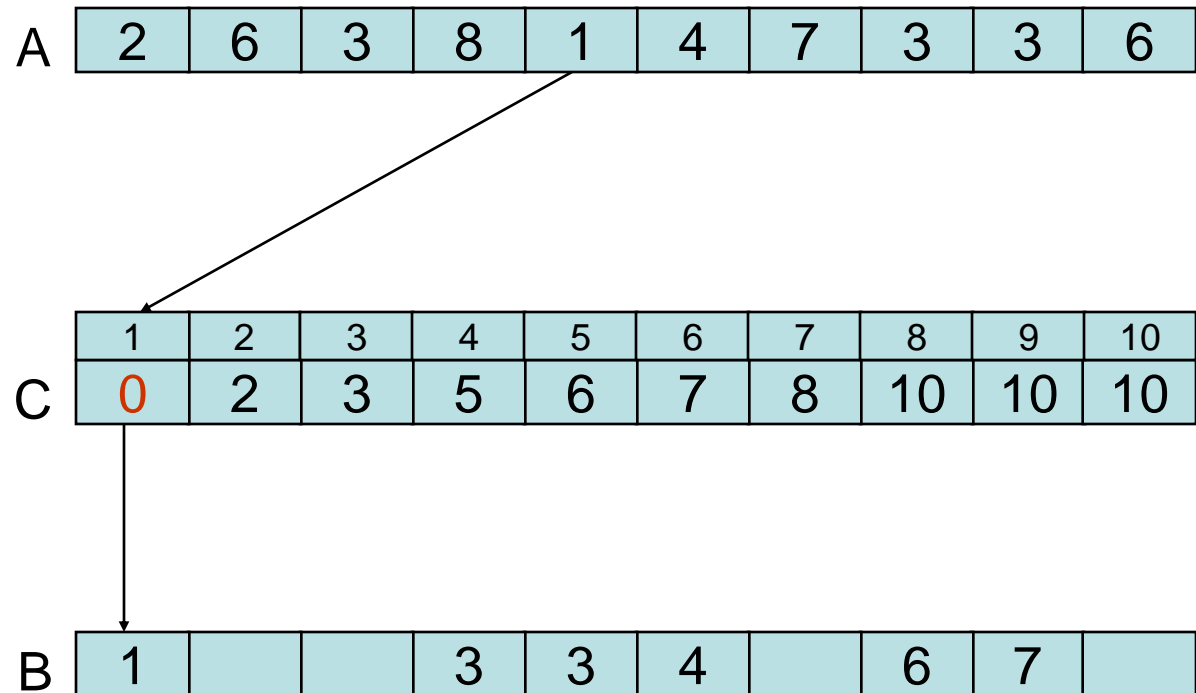
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (35)

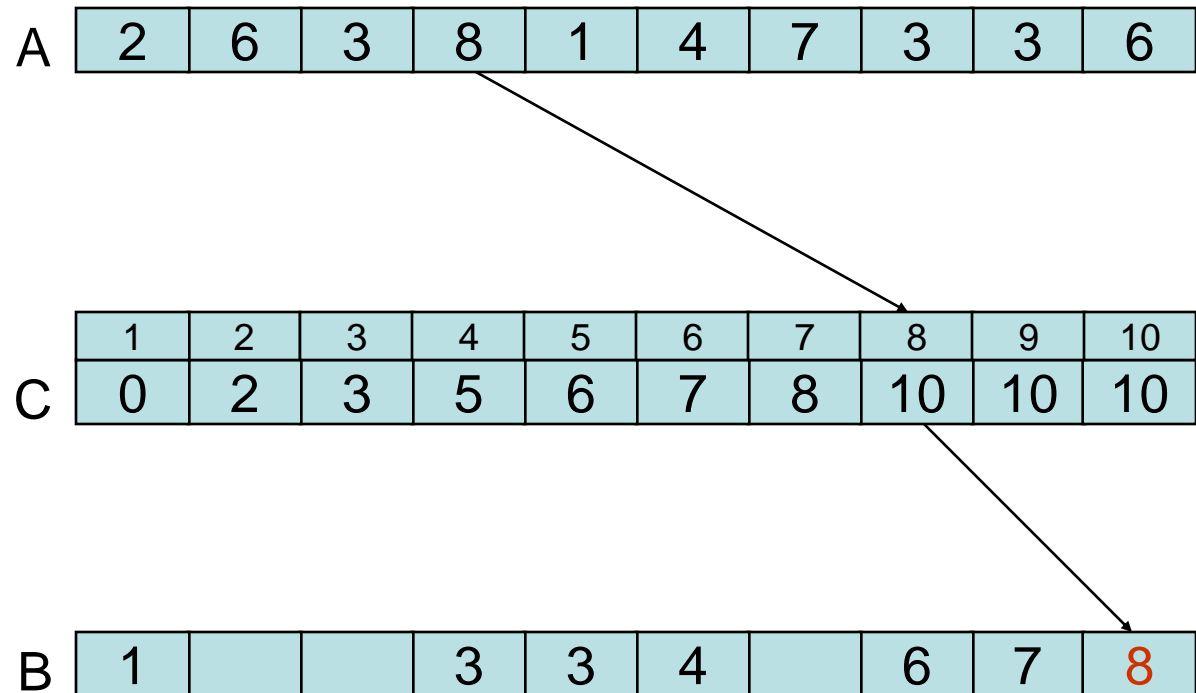
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (36)

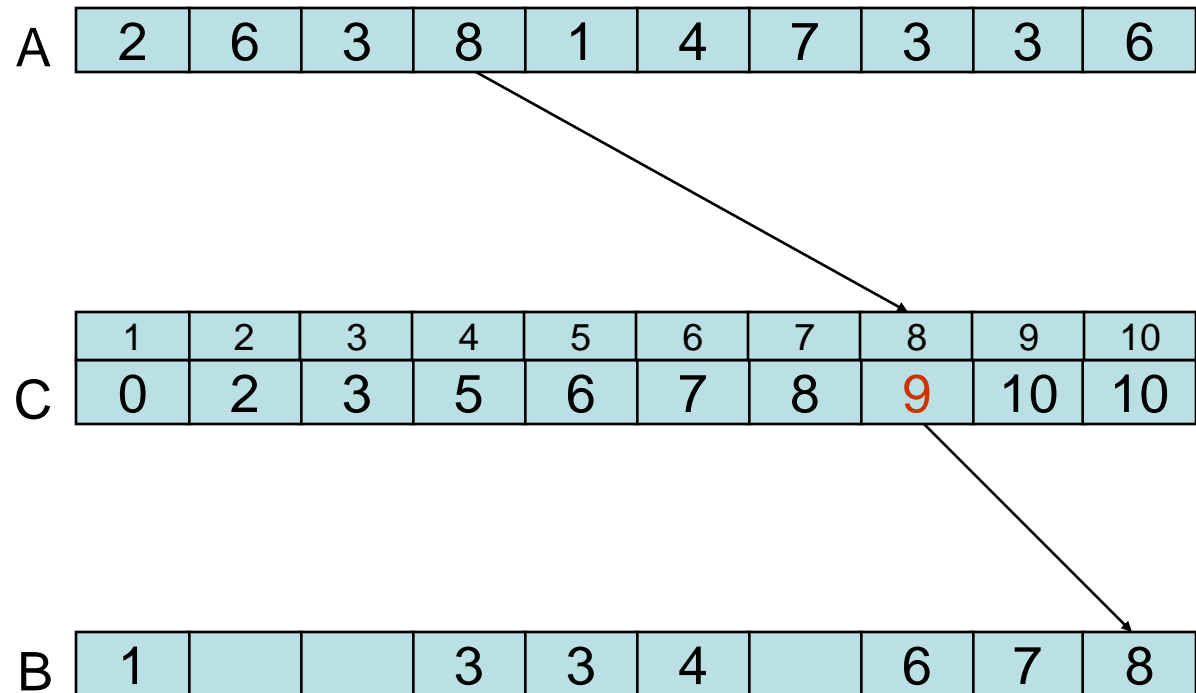
Implementierung



universität
wien

1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



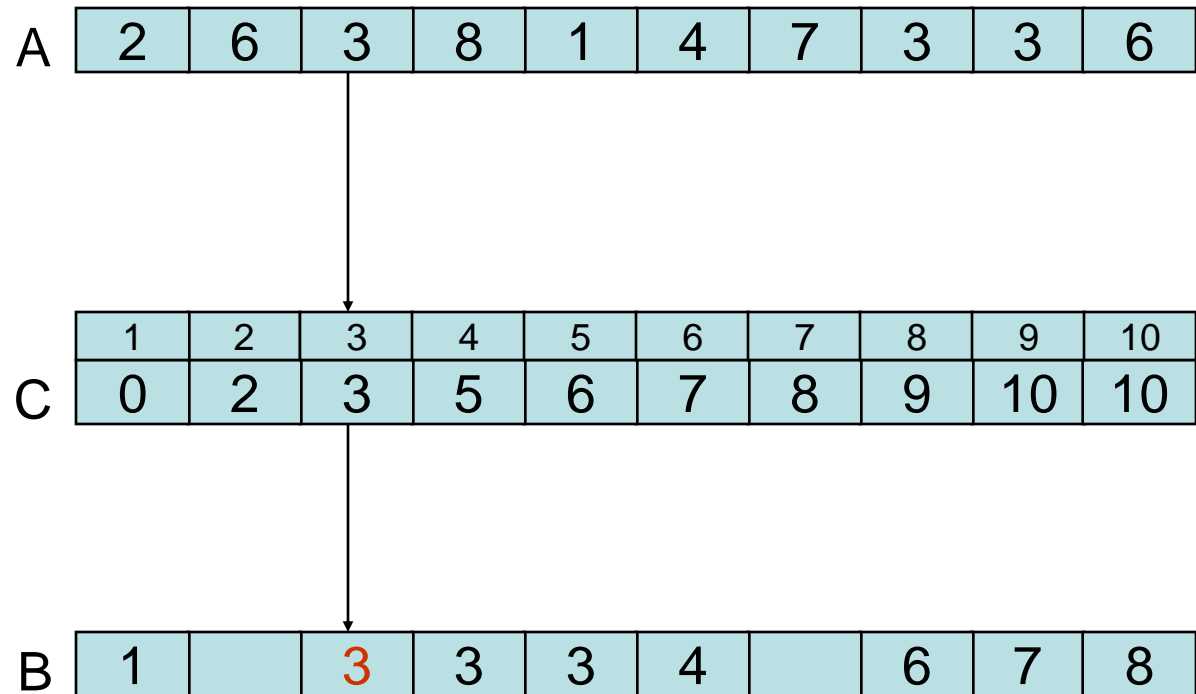
Counting Sort (37)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



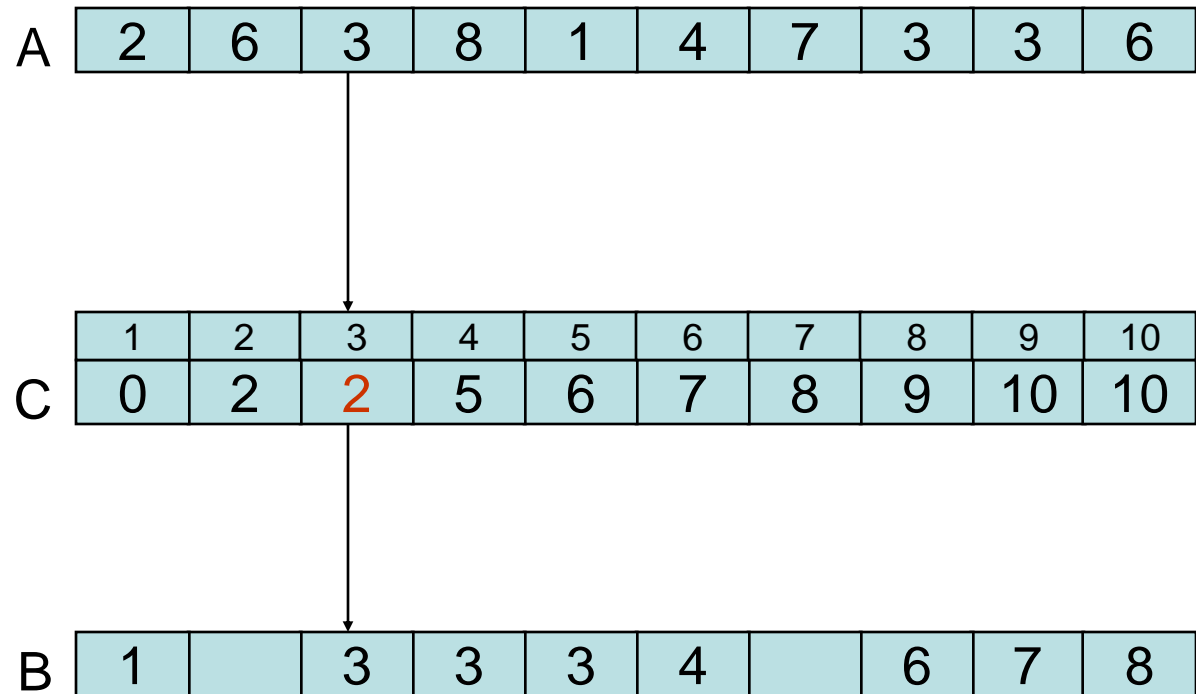
Counting Sort (38)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



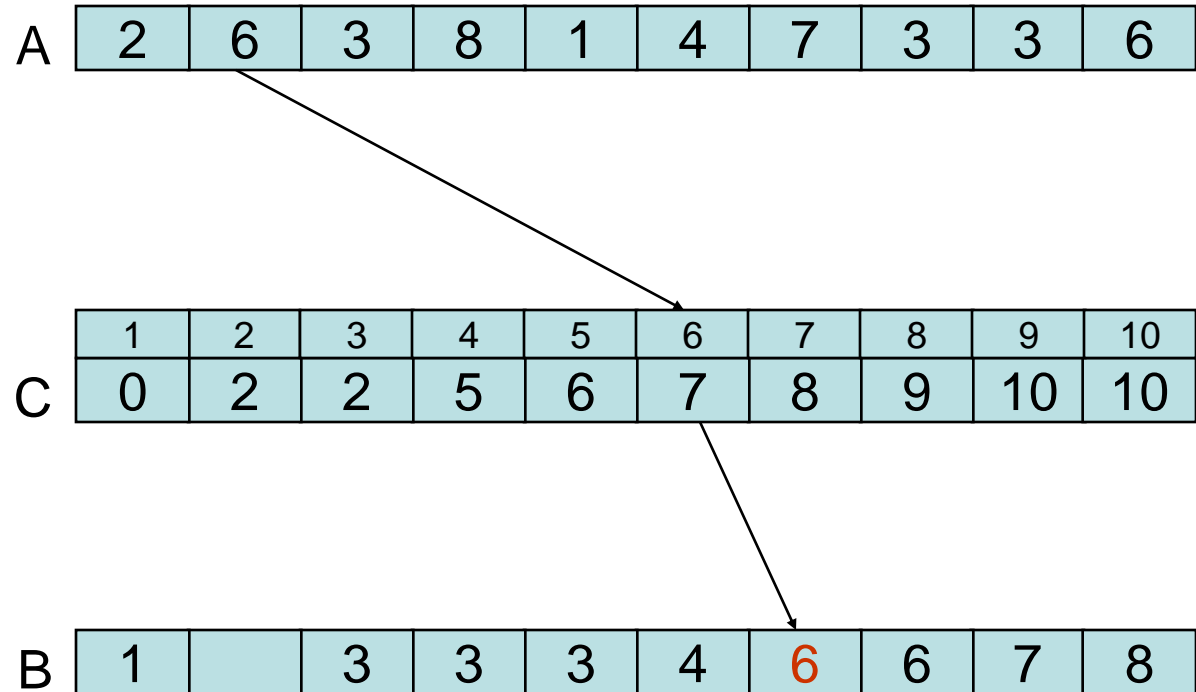
Counting Sort (39)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



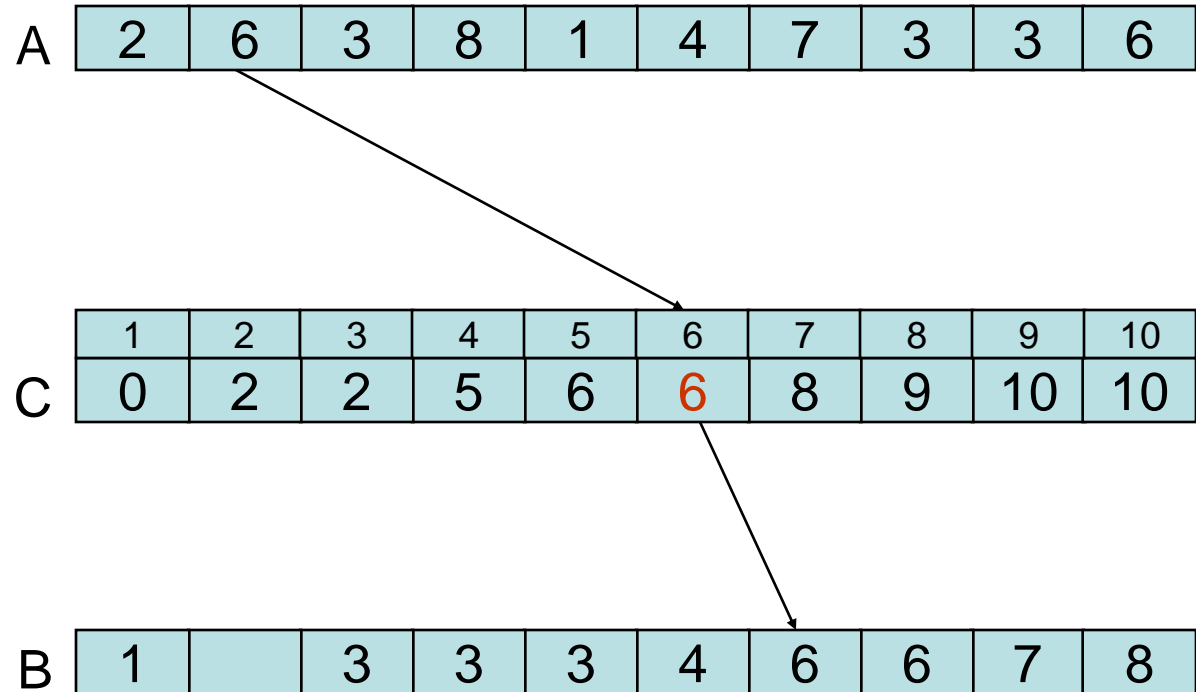
Counting Sort (40)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



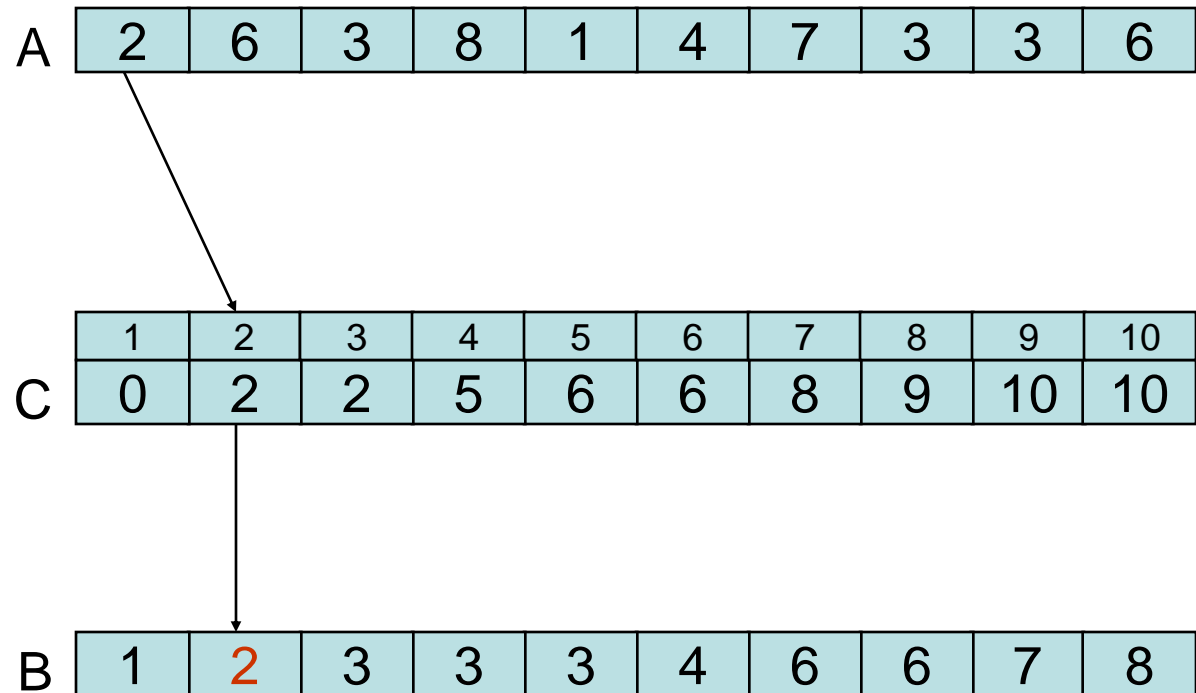
Counting Sort (41)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



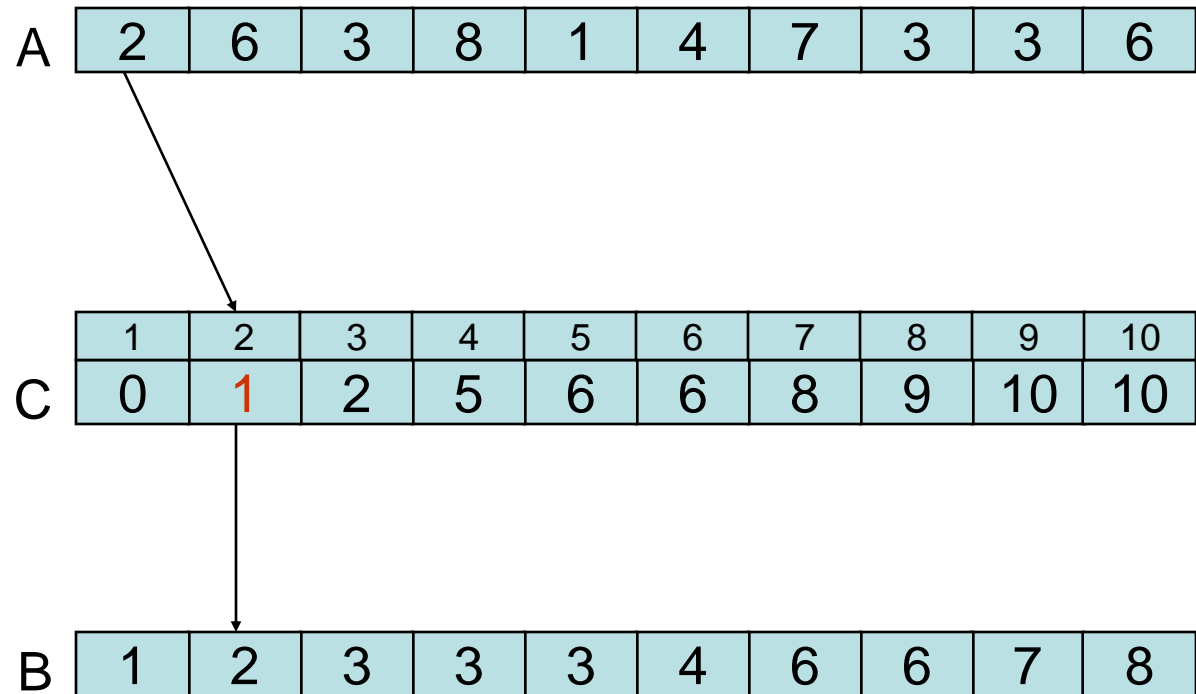
Counting Sort (42)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. **Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]] --$**

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```



Counting Sort (43)

Implementierung



1. Zählen der Vorkommnisse
2. Aufsummieren in C (Anzahl der Elemente die kleiner sind als $A[i]$)
3. Von hinten auslesen, in $B[C[A[i]]]$ einfügen und $C[A[i]]$ --

```
for i = 1 to k
  do C[i] = 0
for j = 1 to length(A)
  do C[A[j]] = C[A[j]] + 1
for i = 2 to k
  do C[i] = C[i] + C[i-1]
for j = length[A] downto 1 do
  B[C[A[j]]] = A[j]
  C[A[j]] = C[A[j]] - 1
```

A

2	6	3	8	1	4	7	3	3	6
---	---	---	---	---	---	---	---	---	---

C

1	2	3	4	5	6	7	8	9	10
0	1	2	5	6	6	8	9	10	10

B

1	2	3	3	3	4	6	6	7	8
---	---	---	---	---	---	---	---	---	---

