

יסודות האלגוריתמים והסיבוכיות

תרגול 02 - מיון מנייה



בעיית המיון בבעיית המיון אנו מקבלים מערך A בו n איברים ואנו מתבקשים לבנות ממנו מערך ממויין.



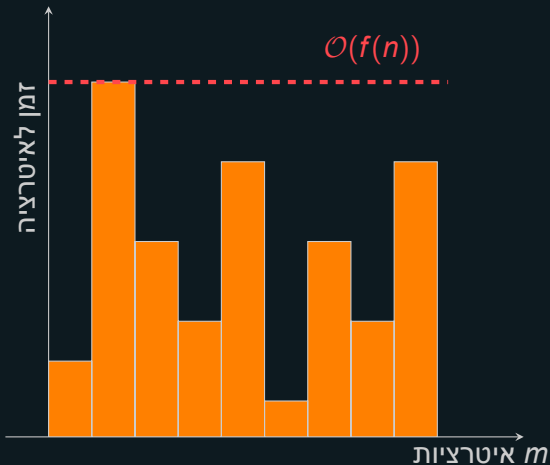
בעיית המיון בבעיית המיון אנו מקבלים מערך A בו n איברים ואנו מתבקשים לבנות ממנו מערך ממויין.

האלגוריתם הנאיבי



בעיית המיון בבעיית המיון אנו מקבלים מערך A בו n איברים ואנו מתבקשים לבנות ממנו מערך ממויין.

האלגוריתם הנאיבי אפשר למיין את מערך A על ידי חיפוש האיבר המינימלי במערך בסיבוכיות $O(n)$, בכל פעם נוציא את האיבר המינימלי מ A ונוסיף אותו למערך הפלט שלנו. אחרי n חזרות על הפעולה הזאת נקבל מערך ממויין.

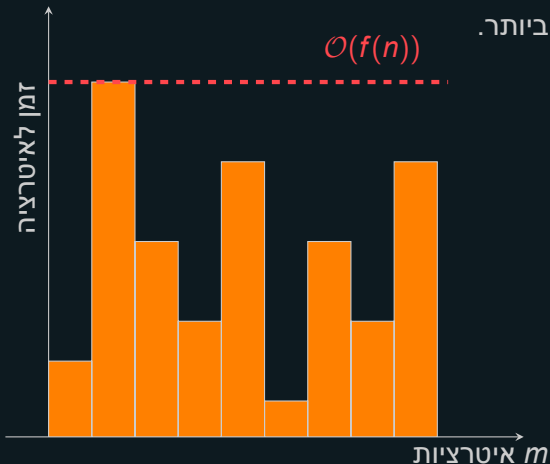




ניתוח לוקאלי

$$\mathcal{O}(m \cdot f_i(n))$$

כש i היא האיטרציה הארוכה ביותר.





ניתוח זמן לוקאלי וגלובלי

ניתוח לוקאלי

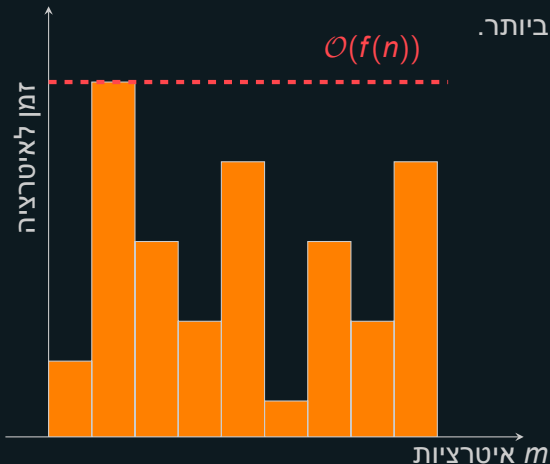
$$\mathcal{O}(m \cdot f_i(n))$$

כש i היא האיטרציה הארוכה ביותר.

ניתוח גלובלי

$$\mathcal{O}\left(\sum_{i \leq m} f_i(n)\right)$$

חסם על סכום האיטרציות.





נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k]$ 
2: Create Array  $B[1, \dots, n]$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k]$ 
2: Create Array  $B[1, \dots, n]$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k]$ 
2: Create Array  $B[1, \dots, n]$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:      $C[A[i]] = C[A[i]] + 1$ 
6:      $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:     while  $C[j] > 0$  do
10:         $B[i] = j$ 
11:         $C[j] = C[j] - 1$ 
12:         $i = i + 1$ 
13:     $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$

C =

--	--	--	--	--	--	--	--

 $B =$

--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k]$ 
2: Create Array  $B[1, \dots, n]$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:      $C[A[i]] = C[A[i]] + 1$ 
6:      $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:     while  $C[j] > 0$  do
10:         $B[i] = j$ 
11:         $C[j] = C[j] - 1$ 
12:         $i = i + 1$ 
13:     $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$

C =

--	--	--	--	--	--	--	--

 $B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k]$   $\triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n]$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:      $C[A[i]] = C[A[i]] + 1$ 
6:      $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:     while  $C[j] > 0$  do
10:         $B[i] = j$ 
11:         $C[j] = C[j] - 1$ 
12:         $i = i + 1$ 
13:     $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$
[illegible] $B =$

--	--	--	--	--	--	--	--



```
14: return  $B[1, \dots, n]$ 
```

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:      $C[A[i]] = C[A[i]] + 1$ 
6:      $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:     while  $C[j] > 0$  do
10:          $B[i] = j$ 
11:          $C[j] = C[j] - 1$ 
12:          $i = i + 1$ 
13:      $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$

C =

--	--	--	--	--	--	--	--

 $B =$

--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

```

1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$

C =

--	--	--	--	--	--	--	--

 $B =$

--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
		1				1	

$C =$

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
		1	1			1	

$C =$

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1		1	1			1	

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1		1	1	1		1	

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1		1	1	1		1	1

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1	1	1	1	1		1	1

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1	1	1	1	1		2	1

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
1	2	1	1	1		2	1

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

--	--	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1							
---	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

	1	2	3	4	5	6	7	8
$C =$	0	1	1	1	1	0	2	1

$B =$

1	2							
---	---	--	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

	1	2	3	4	5	6	7	8
$C =$	0	0	1	1	1	0	2	1

$B =$

1	2	2						
---	---	---	--	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3					
---	---	---	---	--	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4				
---	---	---	---	---	--	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$B =$

1	2	2	3	4	5			
---	---	---	---	---	---	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$B =$

1	2	2	3	4	5			
---	---	---	---	---	---	--	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7		
---	---	---	---	---	---	---	--	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	
---	---	---	---	---	---	---	---	--



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

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

$C =$

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע...!

אולי אפשר לעשות ניתוח יותר חכם ... !



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[j] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע...!

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n
איברים למערך B .



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[j] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע!!!

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n
איברים למערך B .

אנחנו גם יודעים שאנחנו עוברים בסה"כ על
 k תאים של מערך C .



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[j] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

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

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע...!

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n
איברים למערך B .

אנחנו גם יודעים שאנחנו עוברים בסה"כ על
 k תאים של מערך C .



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

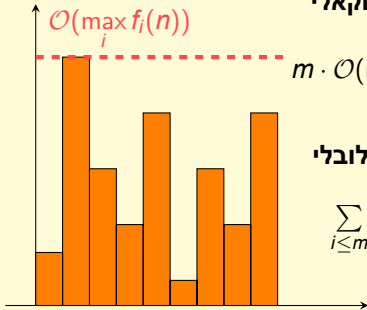
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(1)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[j] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



ניתוח לוקאלי

ניתוח גלובלי

$$\sum_{i \leq m} f_i(n)$$

m איטרציות



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

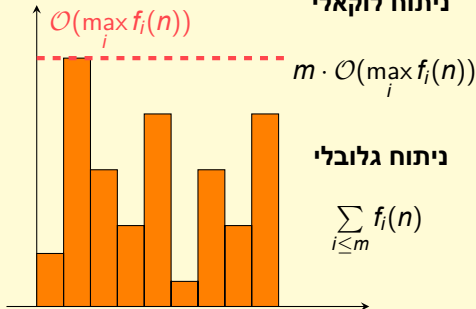
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(1)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[j] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



k איטרציות



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

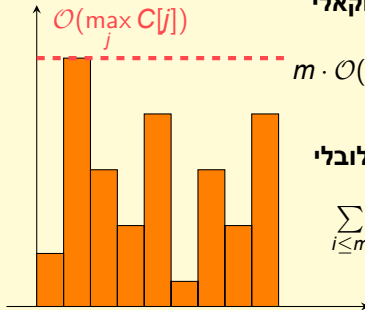
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



ניתוח לוקאלי

$$m \cdot \mathcal{O}(\max_i f_i(n))$$

ניתוח גלובלי

$$\sum_{i \leq m} f_i(n)$$

k איטרציות



ניתוח גלובלי!



נתון המערך [7, 3, 4, 1, 5, 8, 2, 7, 2] פרטו את שלבי מיון מניה.

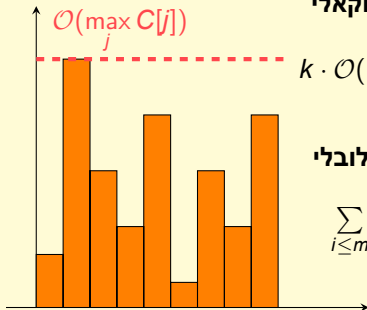
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



ניתוח לוקאלי

ניתוח גלובלי

$$\sum_{i \leq m} f_i(n)$$

k איטרציות



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

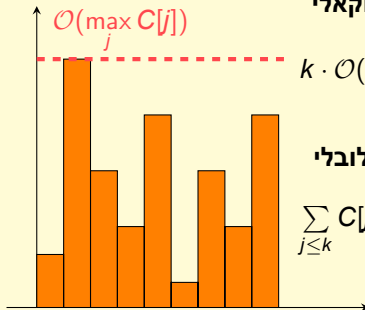
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(1)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



ניתוח לוקאלי

$$k \cdot \mathcal{O}(\max_j C[j])$$

ניתוח גלובלי

$$\sum_{j \leq k} C[j] + \mathcal{O}(1)$$

k איטרציות



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

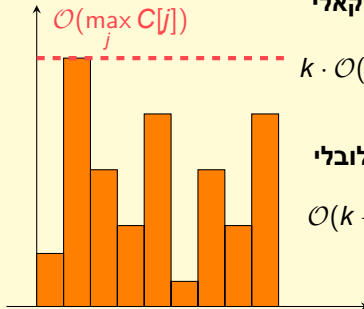
CountSort($A[1, \dots, n], k$)

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright k \cdot \mathcal{O}(n)$ 
9:   while  $C[j] > 0$  do  $\triangleright \mathcal{O}(n)$ 
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

זמן לאיטרציה



ניתוח לוקאלי

$$k \cdot \mathcal{O}(\max_j C[j])$$

ניתוח גלובלי

$$\mathcal{O}(k + \sum_{j \leq k} C[j])$$

k איטרציות



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```
1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright \mathcal{O}(n + k)$ 
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 
```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

1	2	3	4	5	6	7	8

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע!...

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n
איברים למערך B .

אנחנו גם יודעים שאנחנו עוברים בסה"כ על
 k תאים של מערך C .



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright \mathcal{O}(n + k)$ 
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

1	2	3	4	5	6	7	8

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע...!

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n איברים למערך B .

אנחנו גם יודעים שאנחנו עוברים בסה"כ על k תאים של מערך C .



ניתוח גלובלי!



נתון המערך $[7, 3, 4, 1, 5, 8, 2, 7, 2]$ פרטו את שלבי מיון מניה.

CountSort($A[1, \dots, n], k$)

```

1: Create Array  $C[1, \dots, k] \triangleright \mathcal{O}(k)$ 
2: Create Array  $B[1, \dots, n] \triangleright \mathcal{O}(n)$ 
3:  $i = 1$ 
4: while  $i \leq n$  do  $\triangleright \mathcal{O}(n)$ 
5:    $C[A[i]] = C[A[i]] + 1$ 
6:    $i = i + 1$ 
7:  $i = 1$  and  $j = 1$ 
8: while  $j \leq k$  do  $\triangleright \mathcal{O}(n + k)$ 
9:   while  $C[j] > 0$  do
10:     $B[i] = j$ 
11:     $C[j] = C[j] - 1$ 
12:     $i = i + 1$ 
13:    $j = j + 1$ 
14: return  $B[1, \dots, n]$ 

```

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---

$C =$

1	2	3	4	5	6	7	8

$B =$

1	2	2	3	4	5	7	7	8
---	---	---	---	---	---	---	---	---

רגע...!

אולי אפשר לעשות ניתוח יותר חכם ... !

אנחנו הרי יודעים שאנחנו בסה"כ פולטים n איברים למערך B .

אנחנו גם יודעים שאנחנו עוברים בסה"כ על k תאים של מערך C .

זמן ריצה כולל של $\mathcal{O}(n + k)$



ניתוח גלובלי!



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $\mathcal{O}(n + k)$ כך שיהיה ניתן לענות בזמן $\mathcal{O}(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b (או שניהם) אינם בתחום $[1, k]$.



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

(ב) כתבו מה החישוב שיתבצע במקרה בו a או b (או שניהם) אינם בתחום $[1, k]$.

(K)

$$C = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 & 0 & 2 & 1 \end{bmatrix}$$

 $B =$

--	--	--	--	--	--	--	--



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1								



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3							



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4						



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5					



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6				



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6			



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8		



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8	9	



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $[a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8	9	

כעת עבור כל a ו b שנמצאים בתחום $[1, k]$ נחזיר $B[b] - B[a]$.



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $[a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
a									b
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

(1) אם $a < 1$ וגם $b > k$ נחזיר



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $\mathcal{O}(n + k)$ כך שיהיה ניתן לענות בזמן $\mathcal{O}(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $[a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
a									b
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

(1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $\mathcal{O}(n + k)$ כך שיהיה ניתן לענות בזמן $\mathcal{O}(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
	a			b					
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

(1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.

(2) אם $a < 1$ נחזיר



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
	a			b					
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

(1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.

(2) אם $a < 1$ נחזיר $B[b]$.



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
				a				b	
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

- (1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.
- (2) אם $a < 1$ נחזיר $B[b]$.
- (3) אם $b > k$ נחזיר



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
				a				b	
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

- (1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.
- (2) אם $a < 1$ נחזיר $B[b]$.
- (3) אם $b > k$ נחזיר $B[k] - B[a]$.



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

(1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.

(2) אם $a < 1$ נחזיר $B[b]$.

(3) אם $b > k$ נחזיר $B[k] - B[a]$.

(4) אם $b, a > k$ או $b, a < 1$ נחזיר



נתונה קבוצה של n מספרים שלמים בקטע $[1, k]$.

- (א) בצעו על הקבוצה פעולות בזמן $O(n + k)$ כך שיהיה ניתן לענות בזמן $O(1)$ על השאלה "כמה מספרים בקבוצה הנתונה הם בתחום $(a, b]$?"
- (ב) כתבו מה החישוב שיתבצע במקרה בו a או b אינם בתחום $[1, k]$.

(א)

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$B =$	1	3	4	5	6	6	8	9	

(ב) נחלק לארבעה מקרים:

- (1) אם $a < 1$ וגם $b > k$ נחזיר $B[k]$.
- (2) אם $a < 1$ נחזיר $B[b]$.
- (3) אם $b > k$ נחזיר $B[k] - B[a]$.
- (4) אם $b, a > k$ או $b, a < 1$ נחזיר 0.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממשיים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. נבנה מערך A בגודל m ונכניס אליו את ערכי m המספרים.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. נבנה מערך A בגודל m ונכניס אליו את ערכי m המספרים.

2. נבנה מערך עזר בגודל 5000.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. נבנה מערך A בגודל m ונכניס אליו את ערכי m המספרים.
2. נבנה מערך עזר בגודל 5000.
3. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. נבנה מערך A בגודל m ונכניס אליו את ערכי m המספרים.
2. נבנה מערך עזר בגודל 5000.
3. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
4. נחזיר את $B[X]$.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. $O(n)$. נבנה מערך A בגודל n ונכניס אליו את ערכי m המספרים.
2. נבנה מערך עזר בגודל 5000.
3. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
4. נחזיר את $B[X]$.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. $\mathcal{O}(n)$. נבנה מערך A בגודל n ונכניס אליו את ערכי m המספרים.
2. $\mathcal{O}(1)$. נבנה מערך עזר בגודל 5000.
3. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
4. נחזיר את $B[X]$.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. $\mathcal{O}(n)$. נבנה מערך A בגודל n ונכניס אליו את ערכי m המספרים.
2. $\mathcal{O}(1)$. נבנה מערך עזר בגודל 5000.
3. $\mathcal{O}(n)$. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
4. נחזיר את $B[X]$.



נתונה קבוצה של m מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. $\mathcal{O}(n)$. נבנה מערך A בגודל n ונכניס אליו את ערכי m המספרים.
2. $\mathcal{O}(1)$. נבנה מערך עזר בגודל 5000.
3. $\mathcal{O}(n)$. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
4. $\mathcal{O}(1)$. נחזיר את $B[X]$.



נתונה קבוצה של n מספרים (שלמים, לא בהכרח שונים ולא בהכרח ממוינים) בטווח של בין 1 ל-5000. כיצד ניתן לקבוע כמה פעמים נמצא ערך X כלשהו בקבוצה הנתונה? פרטו את שלבי האלגוריתם ונתחו את סיבוכיות זמן הריצה.

נשתמש בתכונות מיון מניה על מנת לפתור את הבעיה.

1. $\mathcal{O}(n)$. נבנה מערך A בגודל n ונכניס אליו את ערכי n המספרים.
 2. $\mathcal{O}(1)$. נבנה מערך עזר בגודל 5000.
 3. $\mathcal{O}(n)$. עבור כל תא במערך A נגדיל את התא המתאים במערך B ע"פ: $B[A[i]] = B[A[i]] + 1$.
 4. $\mathcal{O}(1)$. נחזיר את $B[X]$.
- ניתוח זמן ריצה – סך הכול $\mathcal{O}(n) = \mathcal{O}(1) + \mathcal{O}(n) + \mathcal{O}(1) + \mathcal{O}(n)$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממיין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
---	---	---	---	---	---	---	---	---



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---



נתון מערך של m מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

$$A = \begin{bmatrix} 7 & 3 & 4 & 1 & 5 & 8 & 2 & 7 & 2 \end{bmatrix}$$

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

[illegible]



נתון מערך של m מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

8

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$D =$	↓	↓	↓	↓	↓	↓	↓	↓	↓
	8	7							

נבנה מערך מנייה C ומערך רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס j ב C לרשימה מקושרת בתא $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$	7	3	4	1	5	8	2	7	2
	1	2	3	4	5	6	7	8	
$C =$	1	2	1	1	1	0	2	1	
$D =$	↓	↓	↓	↓	↓	↓	↓	↓	↓
	8	7							

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
5	7							
↓								
8								

נבנה מערך מנייה C ומערך רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס j ב C לרשימה מקושרת בתא $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

4 7
↓
5
↓
8

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1	
---	---	---	---	---	---	---	---	--

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

3 7
↓
4
↓
5
↓
8

נבנה מערך מנייה C ומערך
רשימות מקושרות D .

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1	
---	---	---	---	---	---	---	---	--

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

3
↓
4
↓
5
↓
8

2
↓
7

נבנה מערך מנייה C ומערך
רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1	
---	---	---	---	---	---	---	---	--

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

1
↓
3
↓
4
↓
5
↓
8

2
↓
7

נבנה מערך מנייה C ומערך
רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

1
↓
3
↓
4
↓
5
↓
8

2
↓
7

נבנה מערך מנייה C ומערך
רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס
 j ב C לרשימה מקושרת בתא
 $D[C[j]]$. $\mathcal{O}(k)$



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
1	2							
↓	↓							
3	7							
↓								
4								
↓								
5								
↓								
8								

נבנה מערך מנייה C ומערך רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס j ב C לרשימה מקושרת בתא $D[C[j]]$. $\mathcal{O}(k)$

נעבור בסדר עולה על התאים של D , כשנגיע לתא j נפלוט את איברי הרשימה המקושרת של $D[j]$ החל באיבר הראשון, כל אחד j פעמים.



נתון מערך של m מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2
1	2	3	4	5	6	7

$C =$

1	2	1	1	1	0	2
---	---	---	---	---	---	---

$D =$

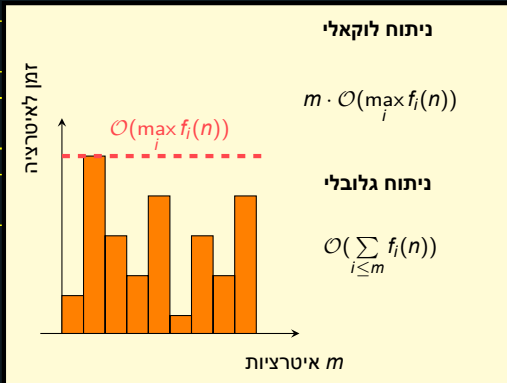
↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---

1
↓
3
↓
4
↓
5
↓
8

2
↓
7

$B =$

1	3	4	5	8	2	2	7	7
---	---	---	---	---	---	---	---	---





נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2
1	2	3	4	5	6	7

$C =$

1	2	1	1	1	0	2
---	---	---	---	---	---	---

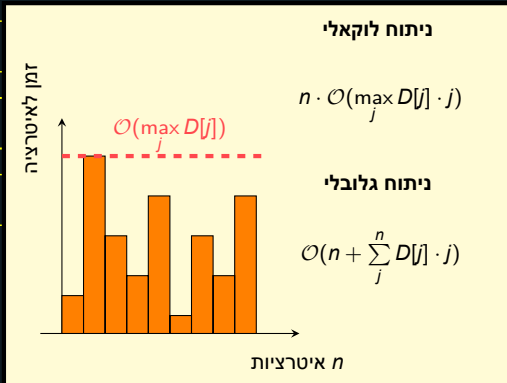
$D =$

↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---

1 ↓ 2
3 ↓ 7
4 ↓
5 ↓
8 ↓

$B =$

1	3	4	5	8	2	2	7	7
---	---	---	---	---	---	---	---	---





נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---

1 2
↓ ↓
3 7
↓
4
↓
5
↓
8

$B =$

1	3	4	5	8	2	2	7	7
---	---	---	---	---	---	---	---	---

נבנה מערך מנייה C ומערך רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס j לרשימה מקושרת בתא $D[C[j]]$. $\mathcal{O}(k)$

נעבור בסדר עולה על התאים של D , כשנגיע לתא j נפלוט את איברי הרשימה המקושרת של $D[j]$ החל באיבר הראשון, כל אחד j פעמים. $\mathcal{O}(n)$

זמן ריצה כולל.



נתון מערך של n מספרים שלמים בתחום 1 עד k . כל מספר עשוי להופיע מספר פעמים במערך. תארו אלגוריתם הממין את המספרים במערך לפי מספר הופעותיהם, מספרים עם מספר הופעות זהה יופיעו במערך הפלט בסדר עולה ע"פ ערכם. כתבו את כל שלבי האלגוריתם ונתחו את זמן ריצת האלגוריתם.

$A =$

7	3	4	1	5	8	2	7	2
1	2	3	4	5	6	7	8	

$C =$

1	2	1	1	1	0	2	1
---	---	---	---	---	---	---	---

$D =$

↓	↓	↓	↓	↓	↓	↓	↓	↓
1	2							
↓	↓							
3	7							
↓								
4								
↓								
5								
↓								
8								

$B =$

1	3	4	5	8	2	2	7	7
---	---	---	---	---	---	---	---	---

נבנה מערך מנייה C ומערך רשימות מקושרות D . $\mathcal{O}(n+k)$

בסדר יורד נוסיף כל אינדקס j לרשימה מקושרת בתא $D[C[j]]$. $\mathcal{O}(k)$

נעבור בסדר עולה על התאים של D , כשנגיע לתא j נפלוט את איברי הרשימה המקושרת של $D[j]$ החל באיבר הראשון, כל אחד j פעמים. $\mathcal{O}(n)$

זמן ריצה כולל. $\mathcal{O}(n+k)$