Politecnico di Torino

Getting Ready for the Exam

Paolo Camurati

Edited by Josie E. Rodriguez

Goal of the Theory exercises in the exam

To assess:

- The understanding of fundamental concepts in computer sciences
- The ability to analyze, evaluate , and face conceptual questions

What do you need:

- The knowledge of:
  - Complexity analysis
  - Algorithms for online connectivity
  - Sorting algorithms
  - Boolean algebra

Topics

- **Union-find algorithms for online connectivity**
  - Quick-find
  - Quick-union
  - Weighted quick-union
- **Quadratic iterative sorting algorithms**
  - Insertion sort
  - Bubble / exchange sort
  - Selection sort
  - Shell sort
- **Linear iterative sorting algorithms**
  - Counting sort
  - Radix sort
- **Linearithmic iterative sorting algorithms**
  - Bottom-up merge sort

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Example:

Given the following function Foo

```
int Foo (int *collector, int n, int k)
{
   int i, j, acc_l;
   for (i=0; i<n-4; i++)
   {
      acc_l = 0;
      for (j=0; j<4; j++)
         acc_l += collector[i+j];
      if (acc_l == k)
         return i;
   }
   return -1;
}
```

Q: Find the worst-case asymptotic complexity (loose bound) of the function $T(n) = O(\ldots)$

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Example:

Given the following function Foo

```
int Foo (int *collector, int n, int k)
{
   int i, j, acc_l;
   for (i=0; i<n-4; i++)
   {
      acc_l = 0;
      for (j=0; j<4; j++)
         acc_l += collector[i+j];
      if (acc_l == k)
         return i;
   }
   return -1;
}
```

Q: Find the worst-case asymptotic complexity (loose bound) of the function T(n) = O(…)

Hint: Observe the loops

Are they nested?

Do they use constant limits?

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Example:

Given the following function Foo

```
int Foo (int *collector, int n, int k)
{
   int i, j, acc_l;
   for (i=0; i<n-4; i++)
   {
      acc_l = 0;
      for (j=0; j<4; j++)
         acc_l += collector[i+j];
      if (acc_l == k)
         return i;
   }
   return -1;
}
```

Q: Find the worst-case asymptotic complexity (loose bound) of the function $T(n) = O(\ldots)$

$$F = (n-4) * (4)$$

$$4(n-4) \quad < \quad n$$

$$T(n) = O(n)$$

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Quick exercise:

Given the following function Foo

```
int Foo (int *collector, int n, int k)
{
   int i, j, acc_l;
   for (i=0; i<n; i++)
   {
      acc_l = 0;
      for (j=0; j<n-3; j++)
         acc_l += collector[i+j];
      if (acc_l == k)
         return i;
   }
   return -1;
}
```

Q: Find the worst-case asymptotic complexity (loose bound) of the function T(n) = O(…)

Structure of the theory exercises

Main target:

Analyze and evaluate (step by step) one numerical exercise with the theoretical concepts

Quick exercise:

```
Void Bar(int A[][n1], int B[][n1])
{
  int C[m1][m1];
  for(int i = 0; i < n1; i++) {
    for(int j = 0; j < n1; j++) {
      C[i][j] = 0;
        for(int k = 0; k < m1; k++) {
          C[i][j] += A[i][k] * B[k][j];
        }
      printf("%d\t", result[i][j]);
   }
   printf("\n");
  }
}
```

Q: Find the worst-case asymptotic complexity (loose bound) of the function T(n) = O(…)

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**. Apply an on-line connectivity algorithm with **quick-find**, returning as a result the content of the array at each step. The **vertices** are named with integers between **0** and **9**.

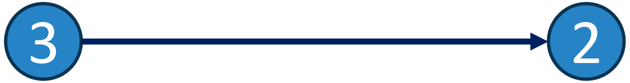
Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**. Apply an on-line connectivity algorithm with **quick-find**, returning as a result the content of the array at each step. The **vertices** are named with integers between **0** and **9**.

3-2



quick-find: updates all array IDs

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

```
for all (p, q):
    read (p, q)
    if (p, q) is connected (id[p] = id[q]),
        do nothing
    else
        Replace id[p]  values with id[q] values
```

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

	0	1	2	3	4	5	6	7	8	9
Array ID	0	1	2	3	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9
3-4	0	1	4	4	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9
3-4	0	1	4	4	4	5	6	7	8	9
5-1	0	1	4	4	4	1	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9
3-4	0	1	4	4	4	5	6	7	8	9
5-1	0	1	4	4	4	1	6	7	8	9
7-3	0	1	4	4	4	1	6	4	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9
3-4	0	1	4	4	4	5	6	7	8	9
5-1	0	1	4	4	4	1	6	7	8	9
7-3	0	1	4	4	4	1	6	4	8	9
5-7	0	4	4	4	4	4	6	4	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

3-2 3-4 5-1 7-3 5-7 9-1

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
3-2	0	1	2	2	4	5	6	7	8	9
3-4	0	1	4	4	4	5	6	7	8	9
5-1	0	1	4	4	4	1	6	7	8	9
7-3	0	1	4	4	4	1	6	4	8	9
5-7	0	4	4	4	4	4	6	4	8	9
9-1	0	4	4	4	4	4	6	4	8	4

Structure of the theory exercises

Quick exercise:

Let the following sequence of pairs be given:

1-2 3-5 0-1 7-4 5-1 9-0 6-5

where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**. Apply an on-line connectivity algorithm with **quick-find**, returning as a result the content of the array after **5 steps**. The **vertices** are named with integers between **0** and **9**.

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**. Apply an on-line connectivity algorithm with **quick-union**, returning as a result the content of the array at each step. The vertices are named with integers between **0** and **9**.

quick-union: Only updates the evaluated array IDs

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

```
for all (p, q):
    read (p, q)
    if(id[p])* = (id[q])*
        do nothing
    else
        id[(id[p])*] = (id[q])* (connect the pair).
```

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

	0	1	2	3	4	5	6	7	8	9
Array ID	0	1	2	3	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9
3-4	0	3	2	4	4	5	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9
3-4	0	3	2	4	4	5	6	7	8	9
5-1	0	3	2	4	4	4	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9
3-4	0	3	2	4	4	5	6	7	8	9
5-1	0	3	2	4	4	4	6	7	8	9
4-0	0	3	2	4	0	4	6	7	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9
3-4	0	3	2	4	4	5	6	7	8	9
5-1	0	3	2	4	4	4	6	7	8	9
4-0	0	3	2	4	0	4	6	7	8	9
7-3	0	3	2	4	0	4	6	0	8	9

Structure of the theory exercises

Let the following sequence of pairs be given:

1-3 3-4 5-1 4-0 7-3 5-7

Array ID	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
1-3	0	3	2	3	4	5	6	7	8	9
3-4	0	3	2	4	4	5	6	7	8	9
5-1	0	3	2	4	4	4	6	7	8	9
4-0	0	3	2	4	0	4	6	7	8	9
7-3	0	3	2	4	0	4	6	0	8	9
5-7	0	3	2	4	0	4	6	0	8	9

Structure of the theory exercises

Quick exercise:
Let the following sequence of pairs be given:

1-2 3-4 0-1 7-3 5-1 9-0 6-5

where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**. Apply an on-line connectivity algorithm with **quick-union**, returning as a result the content of the array after **4 steps**. The vertices are named with integers between **0** and **9**.

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

Show the contents of the array at each step and draw the result as a forest.

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

Show the contents of the array at each step and draw the result as a forest.

Starting state:

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

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

Starting state:

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

10-5

SZ	1	1	1	1	1	2	1	1	1	1	1
Id	0	1	2	3	4	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

Starting state:	**SZ**	1	1	1	1	1	1	1	1	1	1	1
	Id	0	1	2	3	4	5	6	7	8	9	10
10-5	**SZ**	1	1	1	1	1	2	1	1	1	1	1
	Id	0	1	2	3	4	5	6	7	8	9	5
0-3	**SZ**	1	1	1	2	1	2	1	1	1	1	1
	Id	3	1	2	3	4	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

0-3	**SZ**	1	1	1	2	1	2	1	1	1	1	1
	Id	3	1	2	3	4	5	6	7	8	9	5
2-6	**SZ**	1	1	1	2	1	2	2	1	1	1	1
	Id	3	1	6	3	4	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

0-3	**SZ**	1	1	1	2	1	2	1	1	1	1	1
	Id	3	1	2	3	4	5	6	7	8	9	5
2-6	**SZ**	1	1	1	2	1	2	2	1	1	1	1
	Id	3	1	6	3	4	5	6	7	8	9	5
5-4	**SZ**	1	1	1	2	1	3	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

0-3	**SZ**	1	1	1	2	1	2	1	1	1	1	1
	Id	3	1	2	3	4	5	6	7	8	9	5
2-6	**SZ**	1	1	1	2	1	2	2	1	1	1	1
	Id	3	1	6	3	4	5	6	7	8	9	5
5-4	**SZ**	1	1	1	2	1	3	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

5-4	**SZ**	1	1	1	2	1	3	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	9	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

5-4	**SZ**	1	1	1	2	1	3	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	9	5
4-9	**SZ**	1	1	1	2	1	4	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	5	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

5-4	**SZ**	1	1	1	2	1	3	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	9	5
4-9	**SZ**	1	1	1	2	1	4	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	8	5	5
4-8	**SZ**	1	1	1	2	1	5	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	5	5	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

4-8	**SZ**	1	1	1	2	1	5	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	5	5	5
7-9	**SZ**	1	1	1	2	1	6	2	1	1	1	1
	Id	3	1	6	3	5	5	6	5	5	5	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

4-8	**SZ**	1	1	1	2	1	5	2	1	1	1	1
	Id	3	1	6	3	5	5	6	7	5	5	5
7-9	**SZ**	1	1	1	2	1	6	2	1	1	1	1
	Id	3	1	6	3	5	5	6	5	5	5	5
1-10	**SZ**	1	1	1	2	1	7	2	1	1	1	1
	Id	3	5	6	3	5	5	6	5	5	5	5

Structure of the theory exercises

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

1-10	**SZ**	1	1	1	2	1	7	2	1	1	1	1
	Id	3	5	6	3	5	5	6	5	5	5	5

What about the three diagram?

Structure of the theory exercises

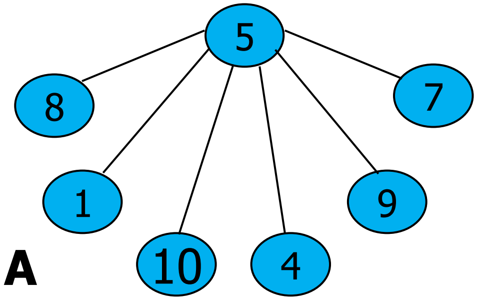
Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

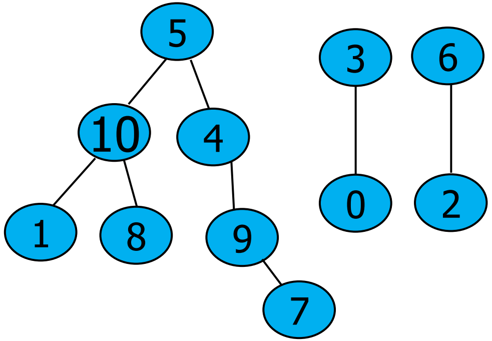
Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

1-10	**SZ**	1	1	1	2	1	7	2	1	1	1	1
	Id	3	5	6	3	5	5	6	5	5	5	5

A or B?



A



B

Structure of the theory exercises

Quick Exercise:

Given the following sequence of pairs, where relation **i-j** indicates that vertex **i** is adjacent to vertex **j**:

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

Apply an on-line connectivity algorithm with **weighted quick union**. Nodes are named with integers between 0 and 10.

Show the contents of the array after **5 steps**.

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Subscripts identify instances of the same key. Show the contents at each step of the intermediate data structures used.

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Subscripts identify instances of the same key. Show the contents at each step of the intermediate data structures used.

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Data are in the range 0..**k**-1.
The largest one is 12 => **k**-1=12, **k**=13

Starting state:

A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	0	0	0	0	0	0	0	0	0	0	0	0	Simple/multiple occurrences array

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

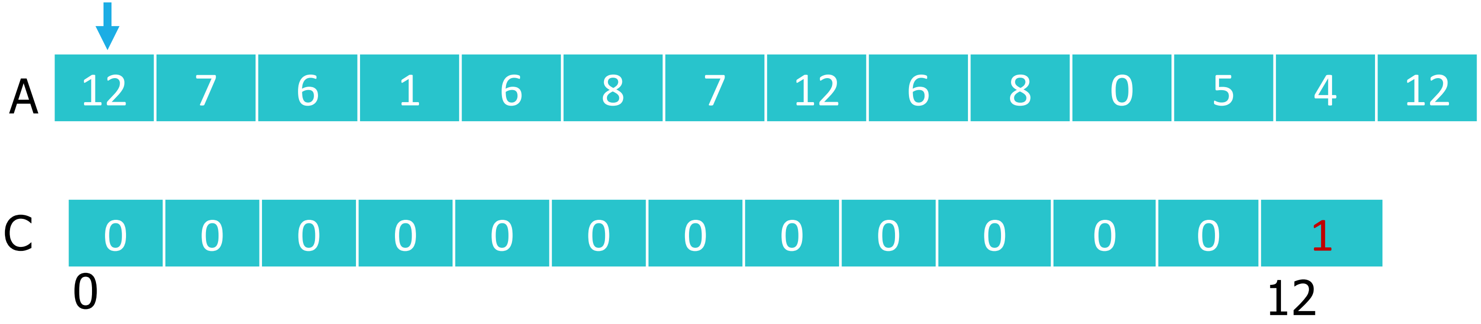
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	0	0	0	0	0	0	0	0	0	0	0	0
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$



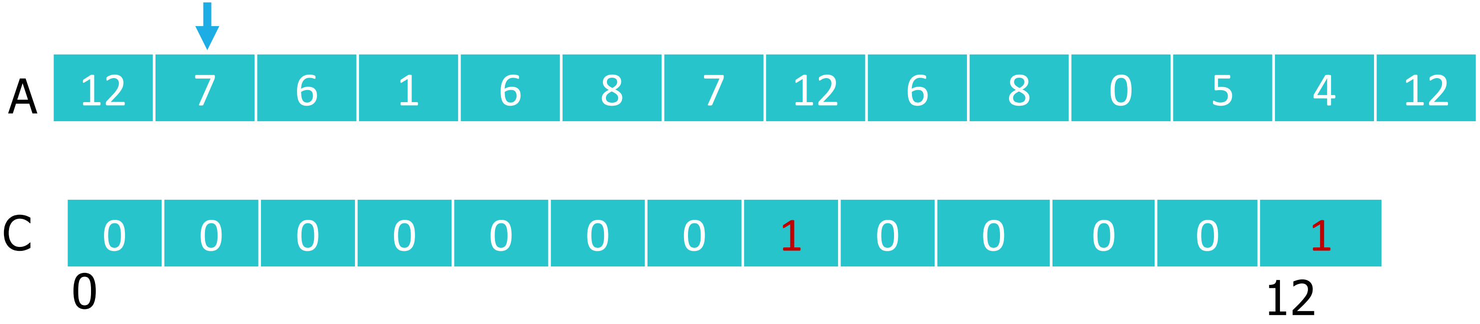
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	0	0	0	0	0	0	0	0	0	0	0	1
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



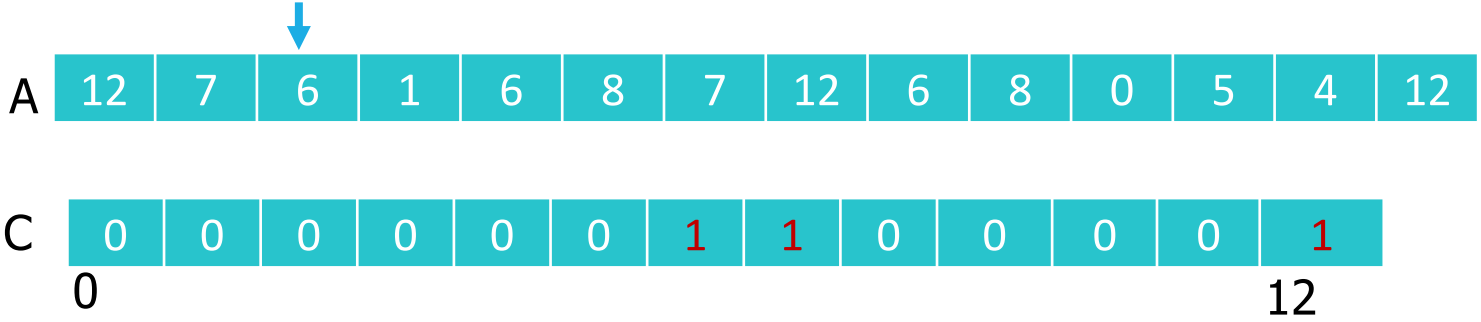
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	0	0	0	0	0	0	1	0	0	0	0	1
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



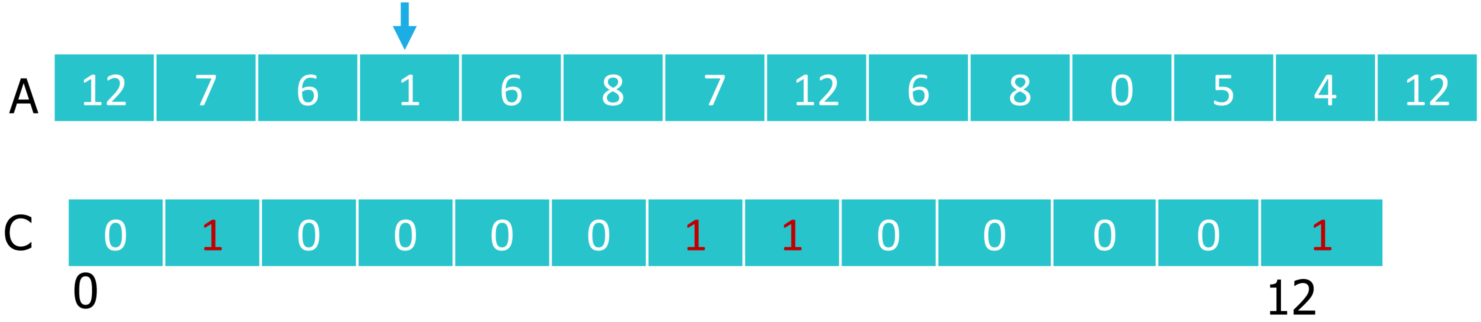
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	0	0	0	0	0	1	1	0	0	0	0	1
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



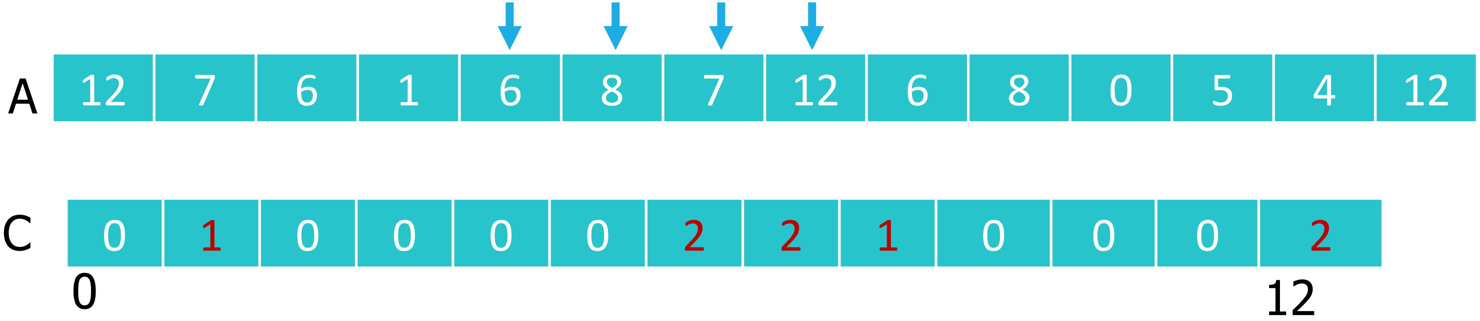
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	0	1	0	0	0	0	1	1	0	0	0	0	1
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



A

12	7	6	1	6	8	7	12	6	8	0	5	4	12

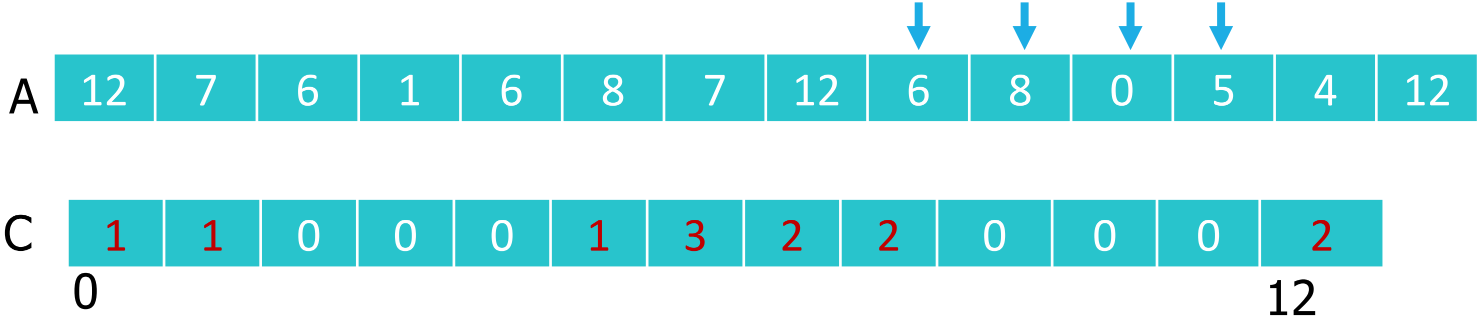
C

0	1	0	0	0	0	2	2	1	0	0	0	2

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



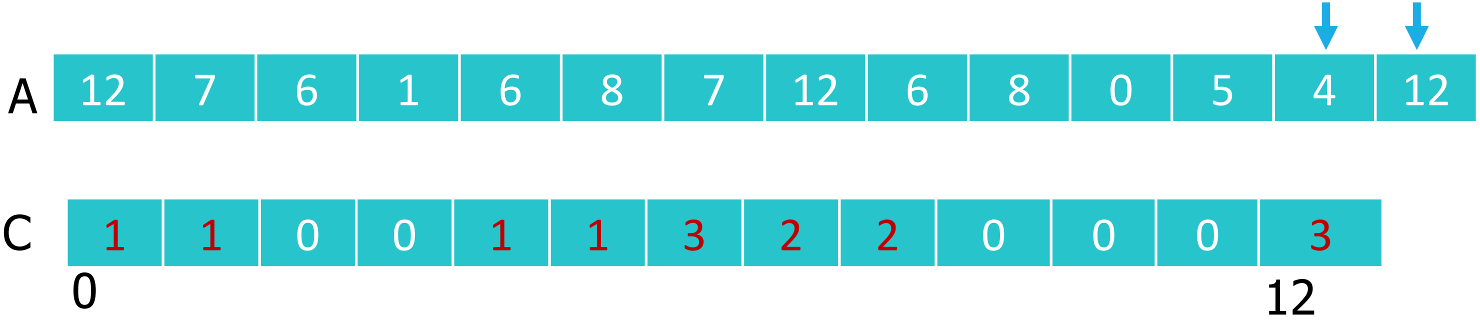
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	1	1	0	0	0	1	3	2	2	0	0	0	2
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3



A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

C	1	1	0	0	1	1	3	2	2	0	0	0	3
	0												12

Simple occurrences

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Multiple occurrences

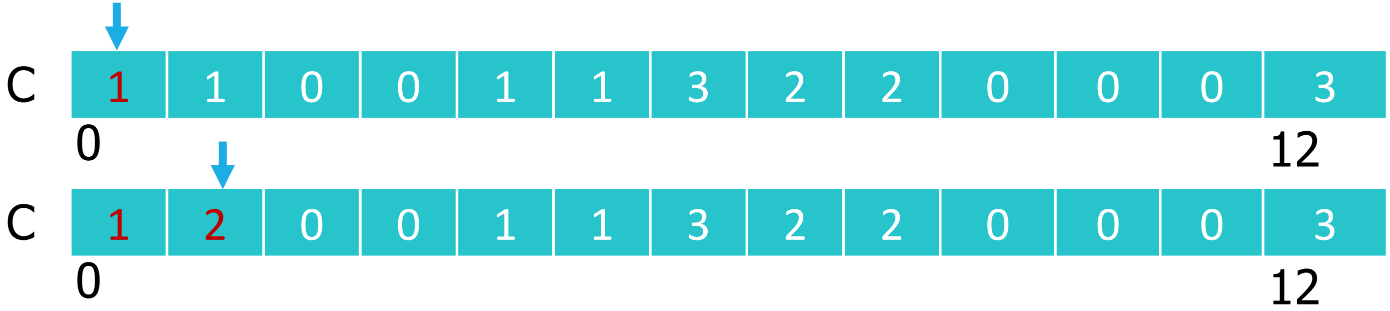
C	1	1	0	0	1	1	3	2	2	0	0	0	3
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Multiple occurrences



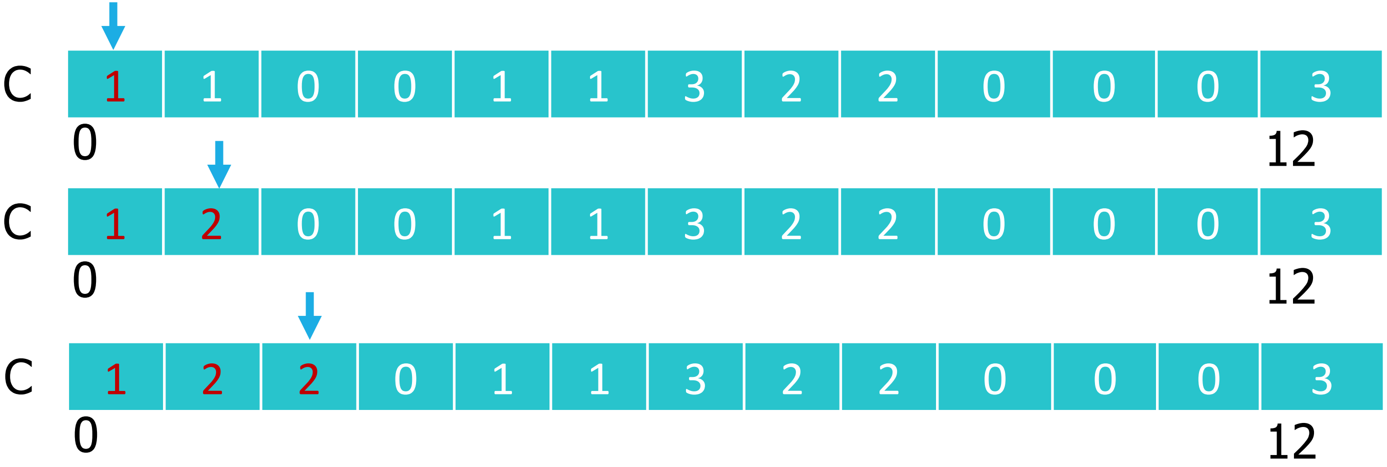
C	1	1	0	0	1	1	3	2	2	0	0	0	3
	0												12
C	1	2	0	0	1	1	3	2	2	0	0	0	3
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Multiple occurrences



C	1	1	0	0	1	1	3	2	2	0	0	0	3
	0												12

C	1	2	0	0	1	1	3	2	2	0	0	0	3
	0												12

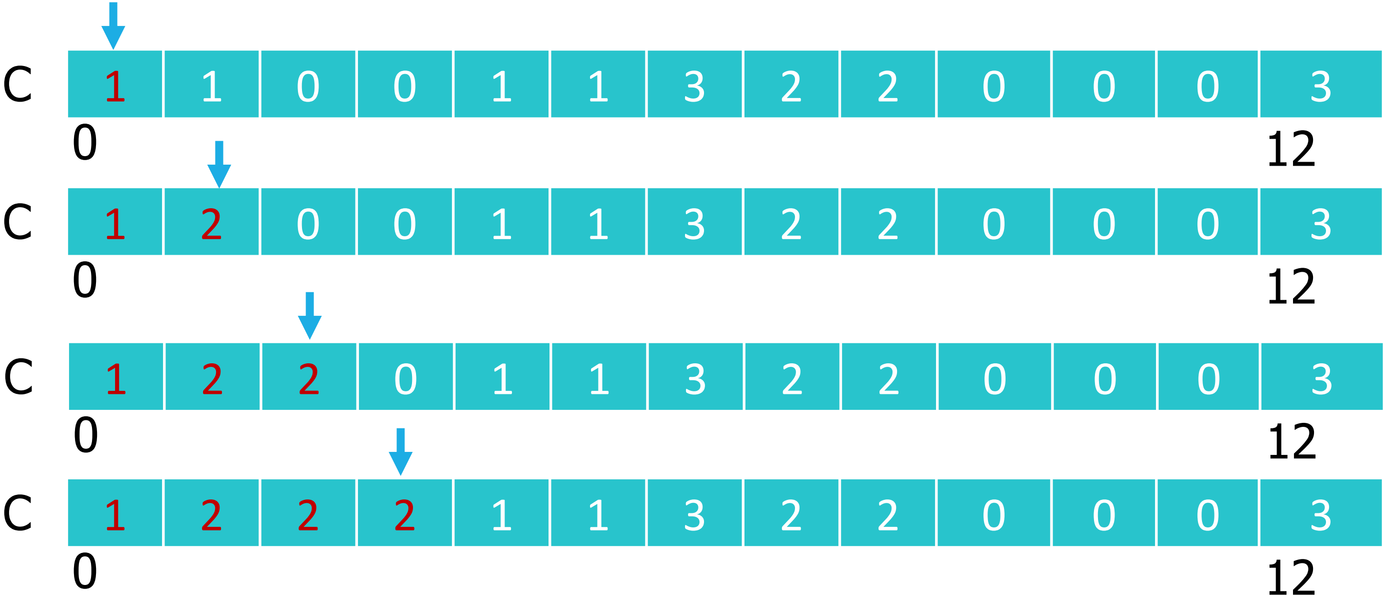
C	1	2	2	0	1	1	3	2	2	0	0	0	3
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Multiple occurrences



C	1	1	0	0	1	1	3	2	2	0	0	0	3
	0												12

C	1	2	0	0	1	1	3	2	2	0	0	0	3
	0												12

C	1	2	2	0	1	1	3	2	2	0	0	0	3
	0												12

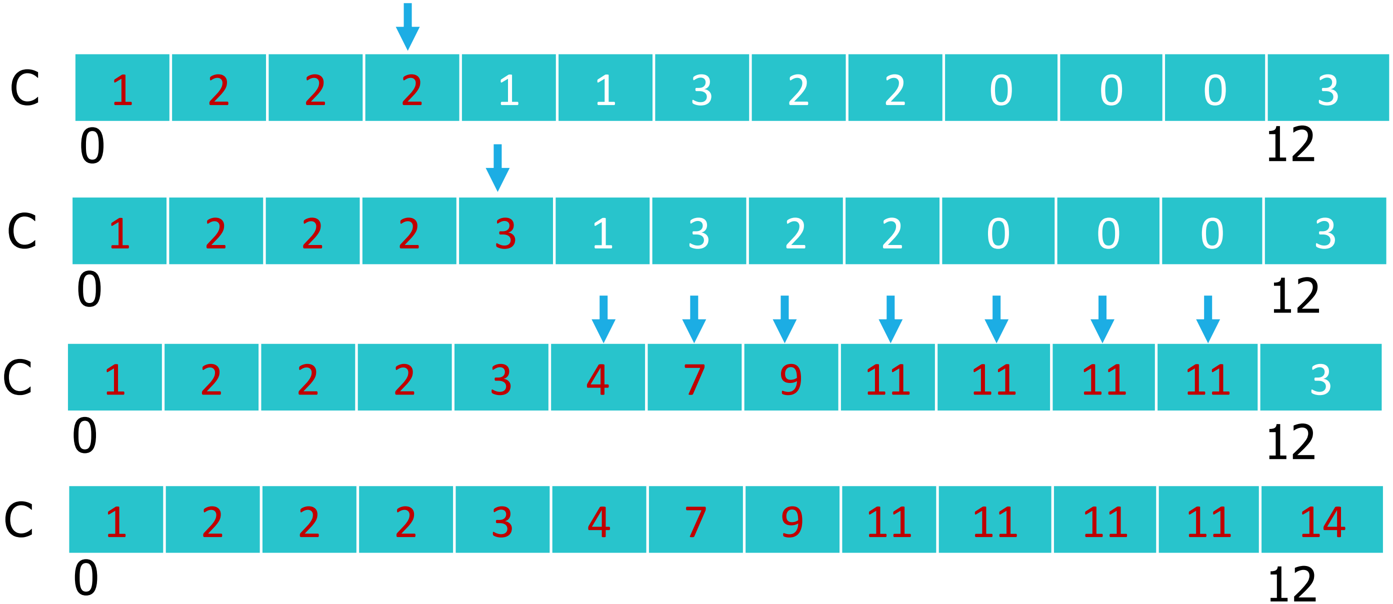
C	1	2	2	2	1	1	3	2	2	0	0	0	3
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Multiple occurrences



C	1	2	2	2	1	1	3	2	2	0	0	0	3
	0												12

C	1	2	2	2	3	1	3	2	2	0	0	0	3
	0												12

C	1	2	2	2	3	4	7	9	11	11	11	11	3
	0												12

C	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result

C	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

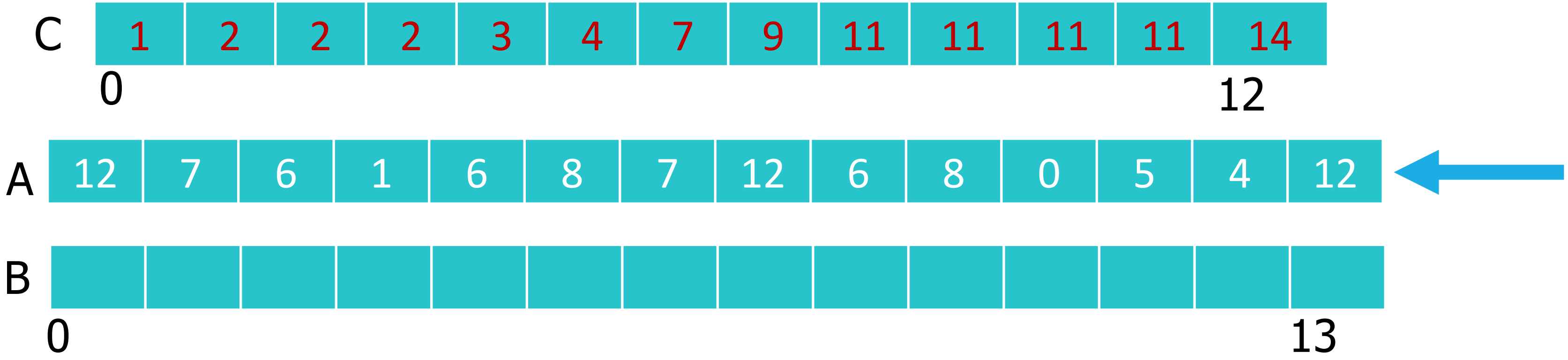
B														

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result



C	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

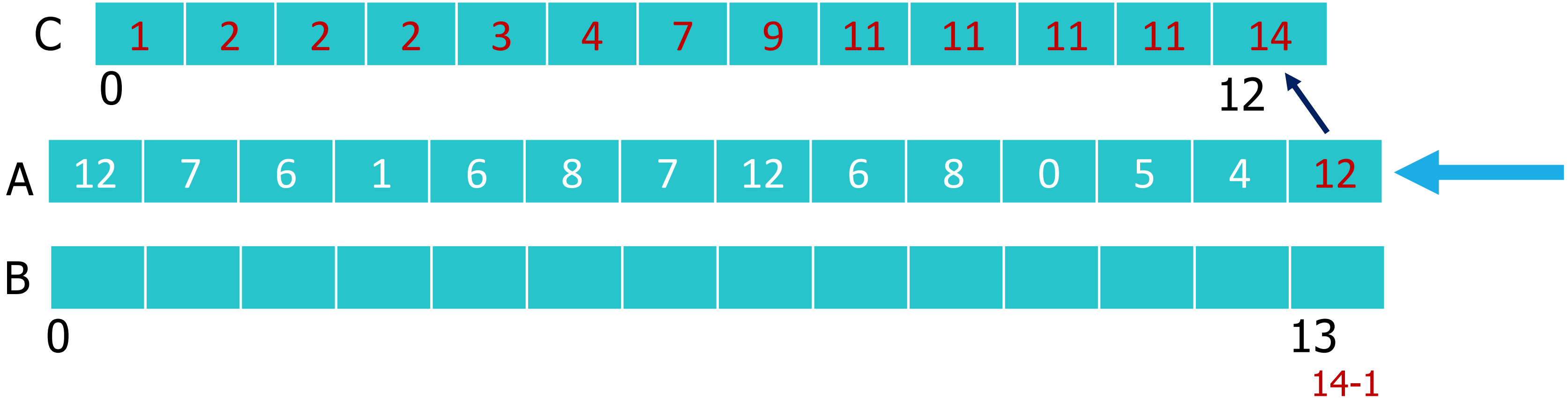
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12
B														
	0													13

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Final result



C													
	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

A														
	12	7	6	1	6	8	7	12	6	8	0	5	4	12

B														
	0													13

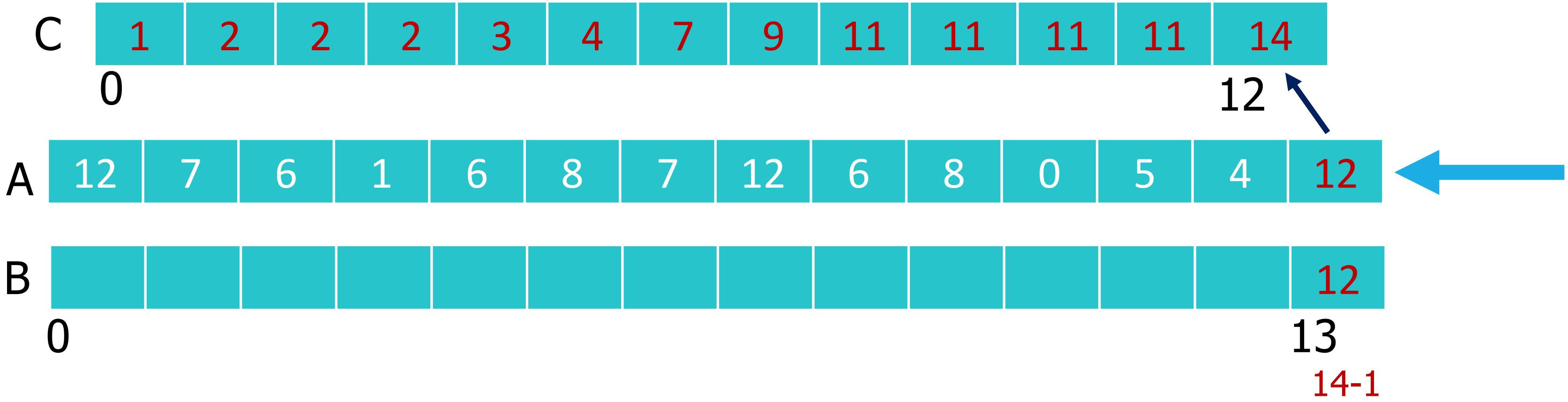
14-1

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Final result



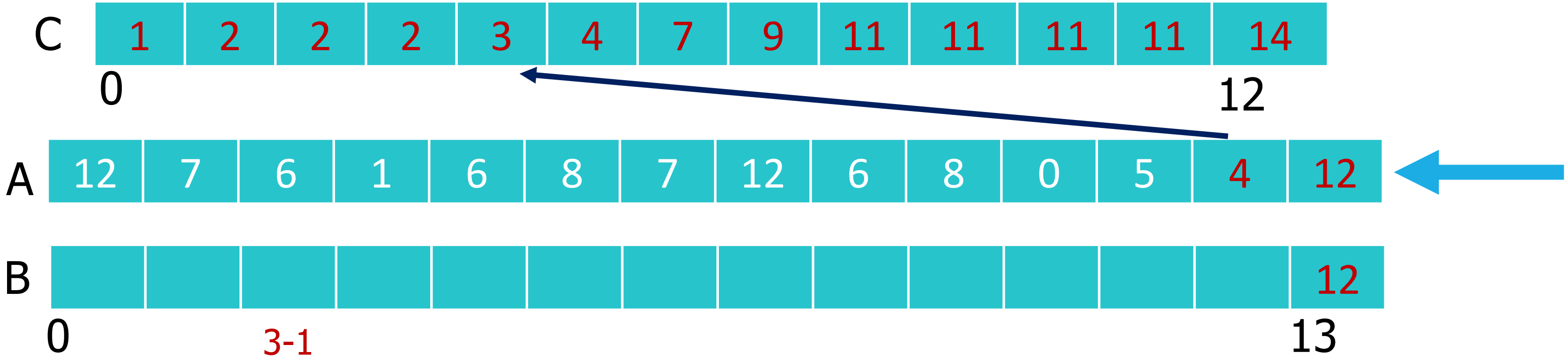
C	1	2	2	2	3	4	7	9	11	11	11	11	14	
	0												12	
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12
B														12
	0													13 14-1

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$$

Final result



C	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

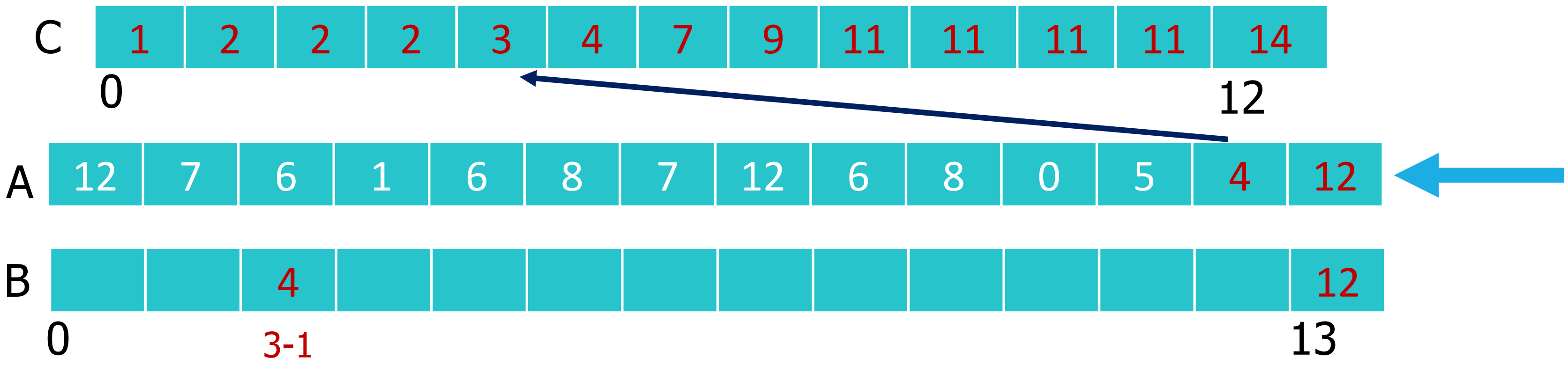
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12
B														12
	0		3-1											13

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result



C

1	2	2	2	3	4	7	9	11	11	11	11	14

A

12	7	6	1	6	8	7	12	6	8	0	5	4	12

B

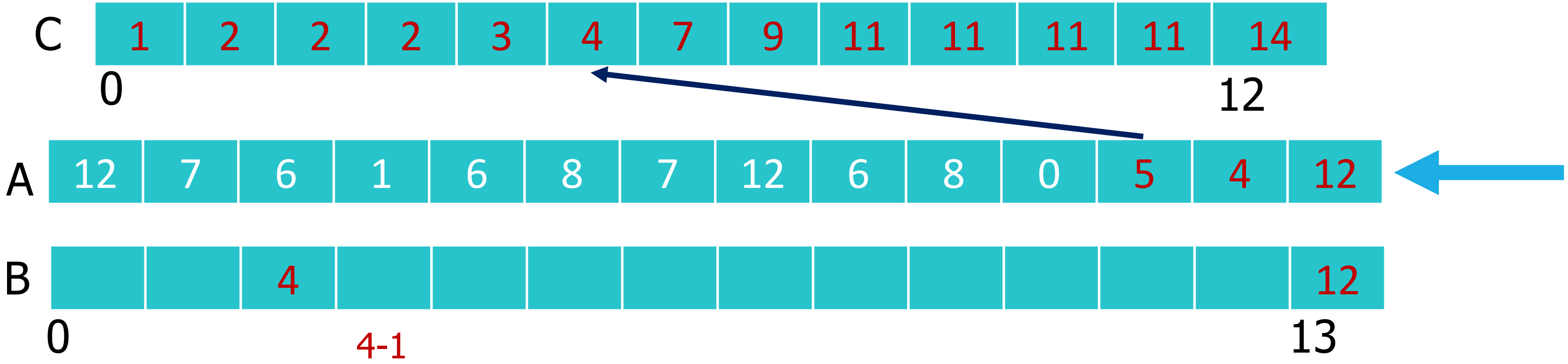
		4											12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Final result



C												
1	2	2	2	3	4	7	9	11	11	11	11	14

A													
12	7	6	1	6	8	7	12	6	8	0	5	4	12

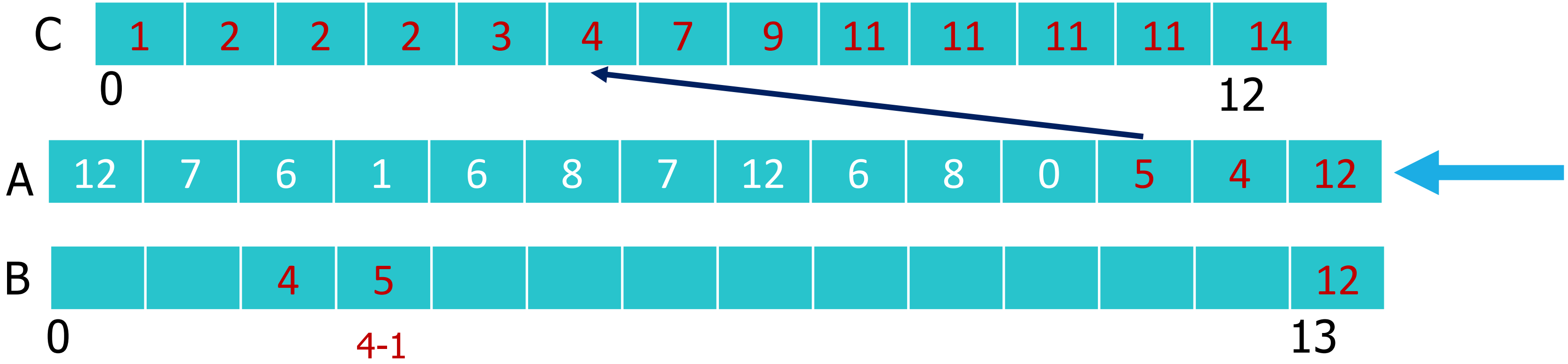
B													
		4											12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

$12_1\ 7_1\ 6_1\ 1\ 6_2\ 8_1\ 7_2\ 12_2\ 6_3\ 8_2\ 0\ 5\ 4\ 12_3$

Final result



C	1	2	2	2	3	4	7	9	11	11	11	11	14
	0												12

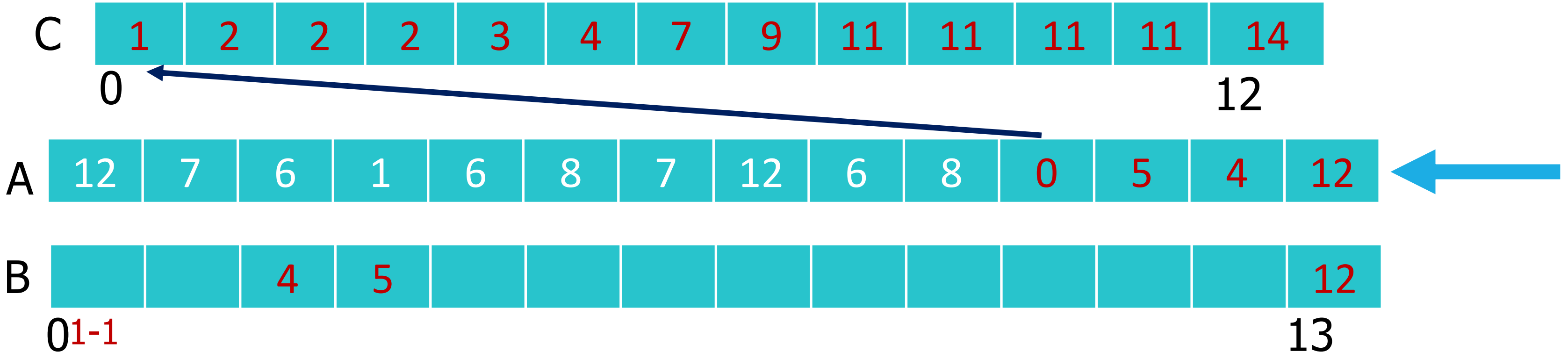
A	12	7	6	1	6	8	7	12	6	8	0	5	4	12
B			4	5										12
	0			4-1										13

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result



C	1	2	2	2	3	4	7	9	11	11	11	11	14

A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

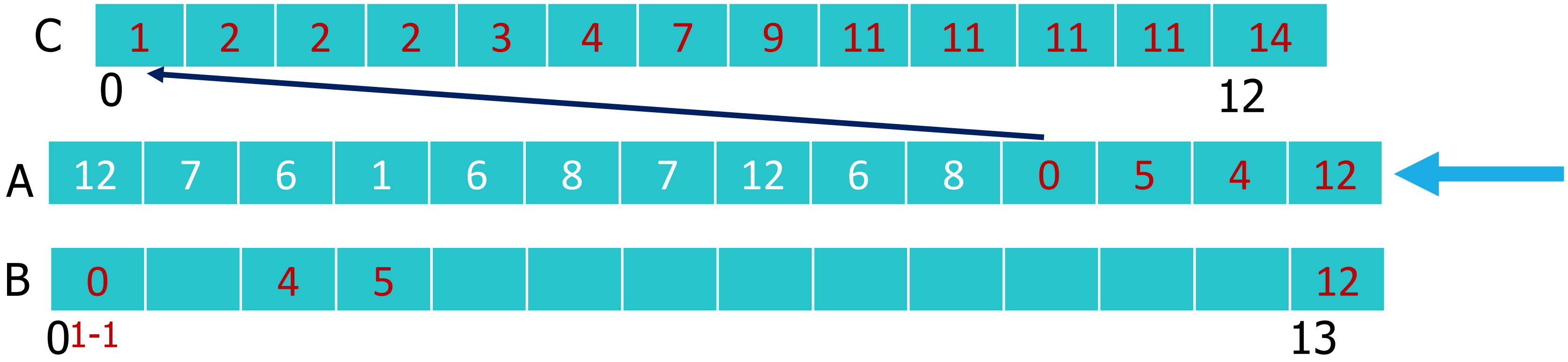
B			4	5										12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result



C	1	2	2	2	3	4	7	9	11	11	11	11	14

A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

B	0		4	5										12

Structure of the theory exercises

Sort the following integer array in ascending order by counting sort:

12_1 7_1 6_1 1 6_2 8_1 7_2 12_2 6_3 8_2 0 5 4 12_3

Final result

	0												12
C	1	2	2	2	3	4	7	9	11	11	11	11	14

A	12	7	6	1	6	8	7	12	6	8	0	5	4	12

	0													13
B	0	1	4	5	6_1	6_2	6_3	7_1	7_2	8_1	8_2	12_1	12_2	12_3