

## Data Structures 2018

### Exercise 10, solutions (Week 46)

1.-2. See file Sort.java.

3.-4. See file Sort.java.

5. Numbers are sorted first based on latter digit:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
-----															
	A1		03		05						BB	0C	FD	FE	AF
			13									CC			
-----															

and then using first digit:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
-----															
03	13									A1	BB	CC	FD		
05										AF				FE	
0C															
-----															

where we obtain the result: 03,05,0C,13,A1,AF,BB,CC,FD,FE.

6. There is, for instance, Shakersort which is modified bubblesort where sorting is bidirectional. This algorithm works efficiently when array is already almost in order, but for this case there are also more efficient algorithms.

7. a)  $A \cup B = \{0, 2, 3, 4, 5, 6\}$  The solution is formed by copying always the smaller of the first elements of sequences A and B to the result sequence and then deleting the copied element. If the first elements are equal, just one of them is copied to result but both are deleted.
- b)  $A \cap B = \{2, 5\}$  The solution is formed by comparing the first elements of sequences A and B and copying one of them to the result sequence if they are equal, both of them are also deleted. If the first elements are not equal the smaller is deleted.
- c)  $A - B = \{0, 3\}$  The solution is formed by comparing the first elements of sequences A and B and copying the smaller to the result sequence, the smaller is also deleted. If the first elements are equal then both are deleted.

$A$	$B$	$a$	$b$	$A \cup B$	$A \cap B$	$A - B$
$\{0, 2, 3, 5\}$	$\{2, 4, 5, 6\}$	0	2	$\{0\}$	$\emptyset$	$\{0\}$
$\{2, 3, 5\}$	$\{2, 4, 5, 6\}$	2	2	$\{0, 2\}$	$\{2\}$	$\{0\}$
$\{3, 5\}$	$\{4, 5, 6\}$	3	4	$\{0, 2, 3\}$	$\{2\}$	$\{0, 3\}$
$\{5\}$	$\{4, 5, 6\}$	5	4	$\{0, 2, 3, 4\}$	$\{2\}$	$\{0, 3\}$
$\{5\}$	$\{5, 6\}$	5	5	$\{0, 2, 3, 4, 5\}$	$\{2, 5\}$	$\{0, 3\}$
$\emptyset$	$\{6\}$	-	6	$\{0, 2, 3, 4, 5, 6\}$	$\{2, 5\}$	$\{0, 3\}$
$\emptyset$	$\emptyset$	-	-	$\{0, 2, 3, 4, 5, 6\}$	$\{2, 5\}$	$\{0, 3\}$

8. The degree of a vertex is the number of edges that come in or go out of the vertex. A self loop is counted twice.

vertex	in	out	deg	vertex	in	out	deg
A	1	2	3	G	2	1	3
B	1	2	3	H	1	2	3
C	1	1	2	I	1	2	3
D	2	2	4	J	2	1	3
E	1	1	2	K	2	1	3
F	2	1	3	L	0	0	0

Graph is not simple since node K has self loop. Graph does not have parallel edges.

