Question 1. Assume that the following figure contains the content of a Sparse Matrix, representation where <line, column, value> triples are kept. How many columns does the Matrix have?

Line	1	1	3	3	4	4
Col	1	7	5	9	4	5
Val	11	9	30	7	20	39

Select one or more:

☐ 6

☐ 9

☐ 5

☐ 10

☒ we cannot determine

We know that there are at least 9 columns, since on column 9 we have a non-zero element. But if there are columns at the end of the matrix with only zero elements, they are not visible on this representation.

The correct answer is: we cannot determine.

Question 2. Assume that the figure below contains a Sparse Matrix in compressed sparse line representation. How many columns are in the matrix?

Line	1	2	3	5	8	10

Col	2	4	3	6	1	2	5	5	8
Val	6	3	3	91	1	3	5	18	102

Select one or more:

☐ 8

☐ 9

☐ 10

☒ we cannot determine

There are at least 8 columns (on column 8 we have a non-zero element), but we do not know how many columns are in total. There might be columns with only zero elements.

The correct answer is: we cannot determine.

Question 3. Assume that the figure contains the elements of a Sparse Matrix, compressed sparse line representation. How many lines are in the matrix?

Line	1	2	3	5	8	10

Col	2	4	3	6	1	2	5	5	8
Val	6	3	3	91	1	3	5	18	102

Select one or more:

☐ 6

☒ 5

☐ 10

☐ we cannot determine

In compressed sparse line representation the line array has always number of lines + 1 elements. Since in the example it contains 6 elements, there are 5 lines in the matrix.

The correct answer is: 5.

Question 4. What is the main difference between ADT Stack and ADT Queue?

Select one or more:

☐ ADT Stack can be implemented on a dynamic array, but ADT Queue cannot

☐ ADT Queue can be implemented on a dynamic array, but ADT Stack cannot

☐ ADT Stack uses two ends of the container, while ADT Queue uses only one end

☒ ADT Stack uses one end of the container, while ADT Queue uses both ends

☐ ADT Stack has iterator, but ADT Queue does not.

☐ ADT Queue has iterator, but ADT Stack does not.

The correct answer is: ADT Stack uses one end of the container, while ADT Queue uses both ends.

Question 5. If we have a fixed-capacity Queue, which operations can throw an exception?

Select one or more:

☒ push

☒ pop

☒ top

☐ isEmpty

☐ isFull

Pop and top can throw an exception for any Queue (fixed capacity or not) if the queue is empty.

But if we have a fixed capacity, we can have a full queue, and then push can also throw an exception.

Technically init can also throw an exception if the capacity is negative. But I did not add this option because this was not discussed separately.

The correct answers are: push, pop, top.

Question 6. What is the main difference between ADT Set and ADT Map?

Select one or more:

☐ In a Set elements are unique, but in a Map we can have the same element multiple times.

☐ In a Map elements are unique, but in a Set we can have the same element multiple times.

☐ A Map can be implemented on a dynamic array, but a Set cannot.

☐ A Set can have an iterator, but a Map cannot.

☒ A Map contains key-value pairs while a Set contains simple elements.

☐ A Set does not have positions, but a Map does.

☐ A Map does not have positions, but a Set has.

ADT Set and ADT Map are petty similar, none of them have positions, both can have iterators, and both contain unique elements. The main difference is that in a Set we have simple elements, while in a Map we have key-value pairs (and actually the keys are unique).

The correct answer is: A Map contains key-value pairs while a Set contains simple elements.

Question 7. Assume that we implement a Queue on a circular dynamic array. Which operation will have a Θ(n) complexity in the worst case?

Select one or more:

☒ push

☐ pop

☐ top

☐ isEmpty

☐ all operations have Θ(1) complexity in the worst case

This is tricky. We use circular arrays to get better complexity for the operations than in case of a regular array. But, it is a dynamic array and we are talking about worst case time complexity and this can happen for push, if we need to do a resize. Since resize happens rarely (assuming correct implementation) push has a Θ(1) amortized complexity.

Optionally, pop could also have Θ(n) worst case performance if we do a resize. But for pop we don't have to do resize.

The correct answer is: push.

Question 8. For ADT Map, what is the parameter for operation search (besides the map), and what does the operation return?

Select one or more:

☐ The parameter is a key and a value and the operations returns true or false depending on whether the pair is in the map or not.

☐ The parameter is a key and the operation returns true or false depending on whether the key is in the map or not.

☒ The parameter is a key and the operation returns the value associated to this key or null_tvalue if the key is not in the map.

☐ The parameter is a value and the operation returns true or false depending on whether this value is in the map or not.

☐ The parameter is a value and the operation returns the key associated to this value or null_tkey if the value is not in the map.

While for most containers search is a boolean operation, here it is different. Operations in a Map happen based on a key. So search receives as parameter the key and returns the associated value.

The correct answer is: The parameter is a key and the operation returns the value associated to this key or null_tvalue if the key is not in the map.