# Question 1:

**Which of the following are true statements?**

#### Select the correct choices:

SubArray and SubString are the same thing, except the former is in the context of any array and the latter is in the context of a String

Both SubArrays and SubStrings are defined to be contiguous group of elements of a given array.

A SubSequence needs to make sure that the order is preserved

SubArray, SubString and SubSequence - all of them preserve order

There is no concept of ordering of elements in a SubSet

# Answer 1:

All the above choices.

# Question 2:

**Is Brute Force Substring Search a linear time algorithm?**

#### Select the correct choice:

Yes. It's complexity is O(Length of Text x Length of Pattern), which is linear

No. It's complexity is O(Length of Text x Length of Pattern), which is not linear time.

Yes. It's complexity is O(Length of Text), which is linear.

Yes, it's complexity is O(Length of Pattern), which is linear.

# Answer 2:

No. It’s complexity is O(Length of Text x Length of Pattern), which is not linear time.

# Question 3:

**In practical implementation, which of the following are two important problems with Brute Force Substring search algorithm?**

#### Select the correct choices:

It's slow. It's not linear time. Linear time is possible and this one doesn't give us that.

It needs us to save previous text. In large corpus of text, when we get a mismatch towards the very end of the pattern, we need to have previous text saved, so that we can go back and increment one.

It uses a lot of memory., because the entire corpus of text needs to be in memory.

When pattern is bigger than text, Brute Force algorithm doesn't work.

# Answer 3:

It’s slow. It’s not linear time. Linear time is possible and this one doesn’t give us that.

It needs us to save previous text. In large corpus of text, when we get a mismatch towards the very end of the pattern, we need to have previous text saved, so that we can go back and increment one.

# Question 4:

**What does the instructor (in the video) say, is a very common problem he sees candidates make in the implementation of brute-force substring search?**

#### Select the correct choice:

They forget to check at the end of the inner loop, if they found a match

They forget to return false at the very end, if there was no match in the entire search

They forget to account for overflow

When there is a mismatch, they forget to reset the index 'i', and instead they restart comparison after the mismatch

# Answer 4:

When there is a mismatch, they forget to reset the index ‘i’ and instead they restart comparison after the mismatch.