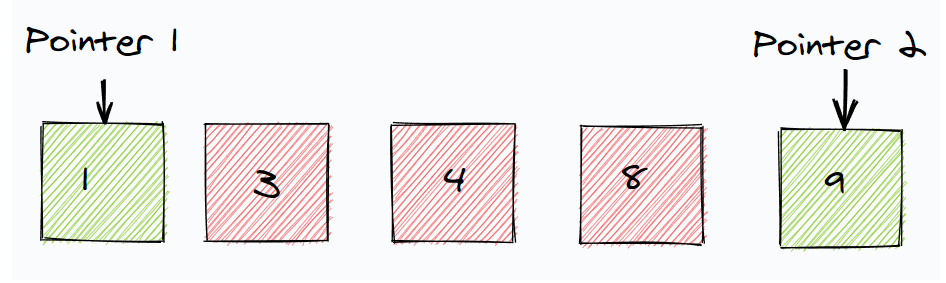
Two Pointer Approach

The two pointer technique is a near necessity in any software developer's toolkit, especially when it comes to technical interviews.

The name two pointers does justice in this case, as it is exactly as it sounds. It's the use of two different pointers (usually to keep track of array or string indices) to solve a problem involving said indices with the benefit of saving time and space.



When do we use it?

In many problems involving collections such as arrays or lists, we have to analyze each element of the collection compared to its other elements.

There are many approaches to solving problems like these. For example we usually start from the first index and iterate through the data structure one or more times depending on how we implement our code.

Sometimes we may even have to create an additional data structure depending on the problem's requirements. This approach might give us the correct result, but it likely won't give us the most space and time efficient result.

This is why the two-pointer technique is efficient. We are able to process two elements per loop instead of just one. Common patterns in the two-pointer approach entail:

Two pointers, each starting from the beginning and the end until they both meet.

One pointer moving at a slow pace, while the other pointer moves at twice the speed.

These patterns can be used for string or array questions. They can also be streamlined and made more efficient by iterating through two parts of an object simultaneously. You can see this in the Two Sum problem or Reverse a String problems.