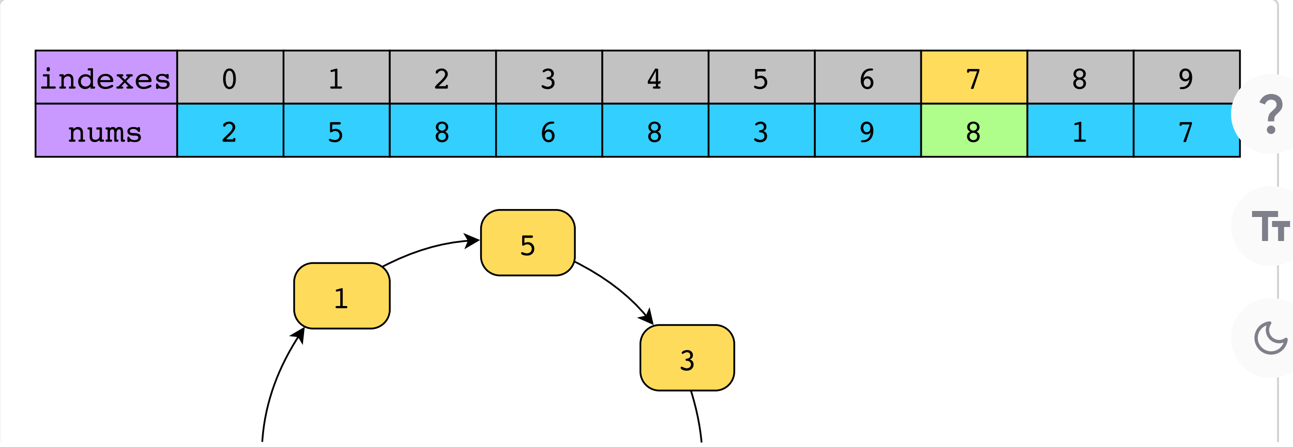
Cyclic Sort: Introduction

Let’s go over the Cyclic Sort pattern, its real-world applications, and some problems we can solve with it.

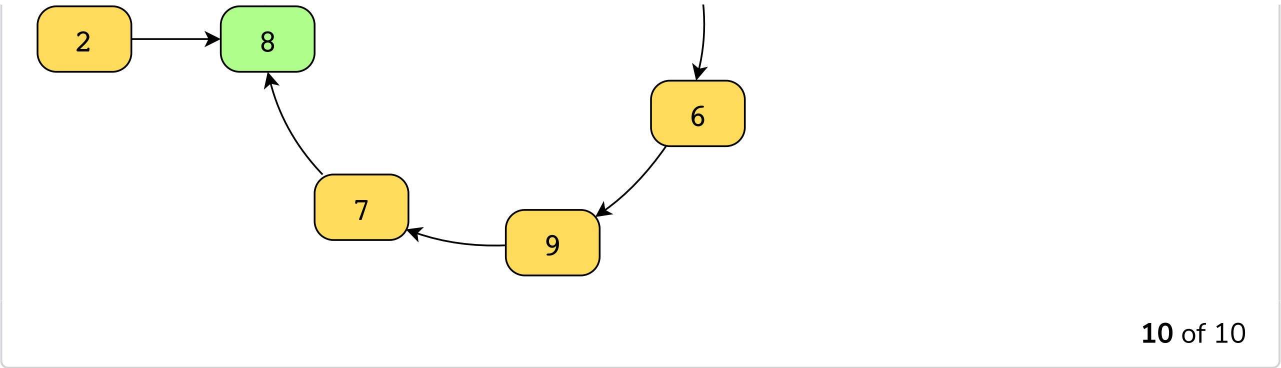
Overview

Cyclic sort is an in-place, unstable, comparison sort algorithm. It is based on the insight that subsequences of numbers in the input array that are not in sorted order actually describe cycles, and that the process of placing each number in its correct position removes these cycles.

The slides below illustrate a traversal through the given array in which the value of the element at the current position determines the next position in the traversal.



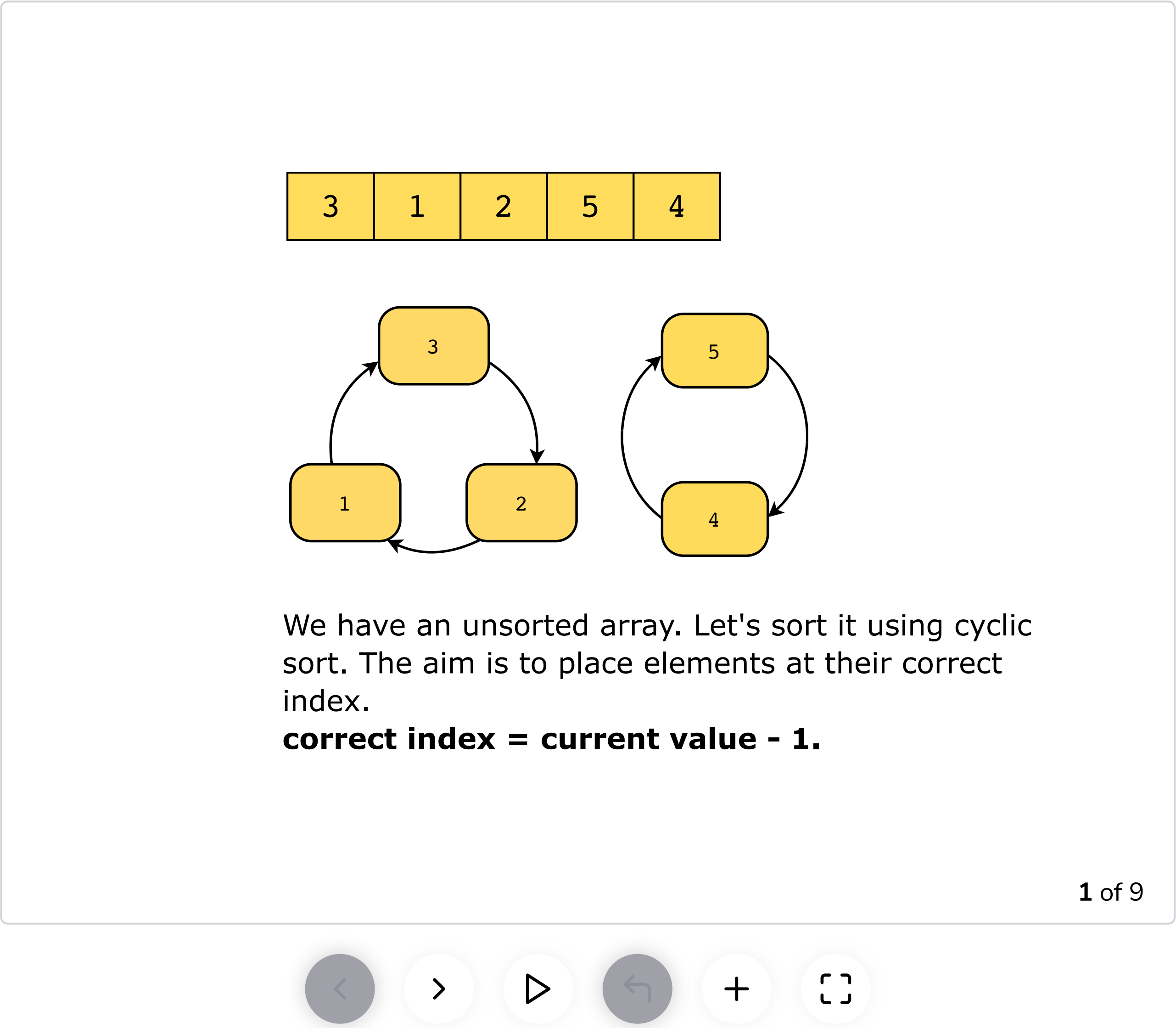
indexes	0	1	2	3	4	5	6	7	8	9
nums	2	5	8	6	8	3	9	8	1	7



As shown in the slides above, we move in a cyclic manner from the 2^{nd} element onwards, returning to the same element after eight moves.

The Cyclic Sort pattern works well for problems involving arrays with numbers ranging from $[1 - n]$, where n is the length of the array, such that one or more cycles exist in it. The algorithm places each element at its correct position within the array. This is achieved by cycling through the array and swapping each element with the element where it should be, then repeating the same operation with the swapped element, continuing in this manner until all the cycles have been removed.

Let's look at an illustration to understand how cyclic sort is used to sort an array.



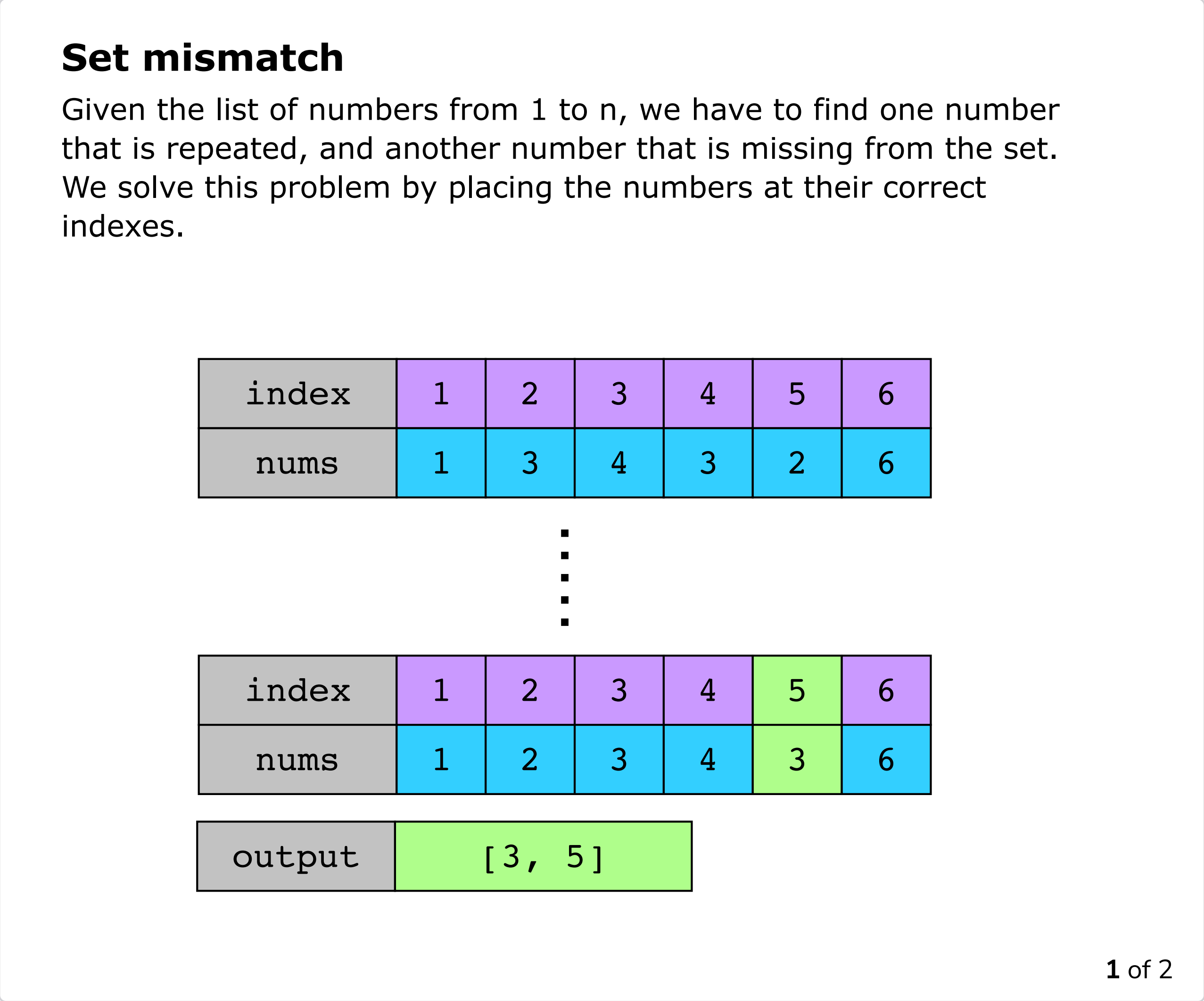
We have an unsorted array. Let's sort it using cyclic sort. The aim is to place elements at their correct index.

correct index = current value - 1.

1 of 9

Examples

The following examples illustrate some problems that can be solved with this approach:



Set mismatch

Given the list of numbers from 1 to n, we have to find one number that is repeated, and another number that is missing from the set. We solve this problem by placing the numbers at their correct indexes.

index	1	2	3	4	5	6
nums	1	3	4	3	2	6

⋮

index	1	2	3	4	5	6
nums	1	2	3	4	3	6

output	[3, 5]

1 of 2

Does my problem match this pattern?

- Yes, if both of these conditions are fulfilled:
  - The problem requires sorting the array without using any additional memory, and the numbers in the array are in the range $[1, n]$, where n is the length of the array.
  - The input array can be divided into cycles.
- No, if any of these conditions is fulfilled:
  - The input data does not contain an array with integer values.
  - The input data is not originally in an array, nor can it be mappe

    - The values in the array are not in the $[1 - n]$ range, where n is the length of the array.
    - The problem requires stable sorting.

Real-world problems

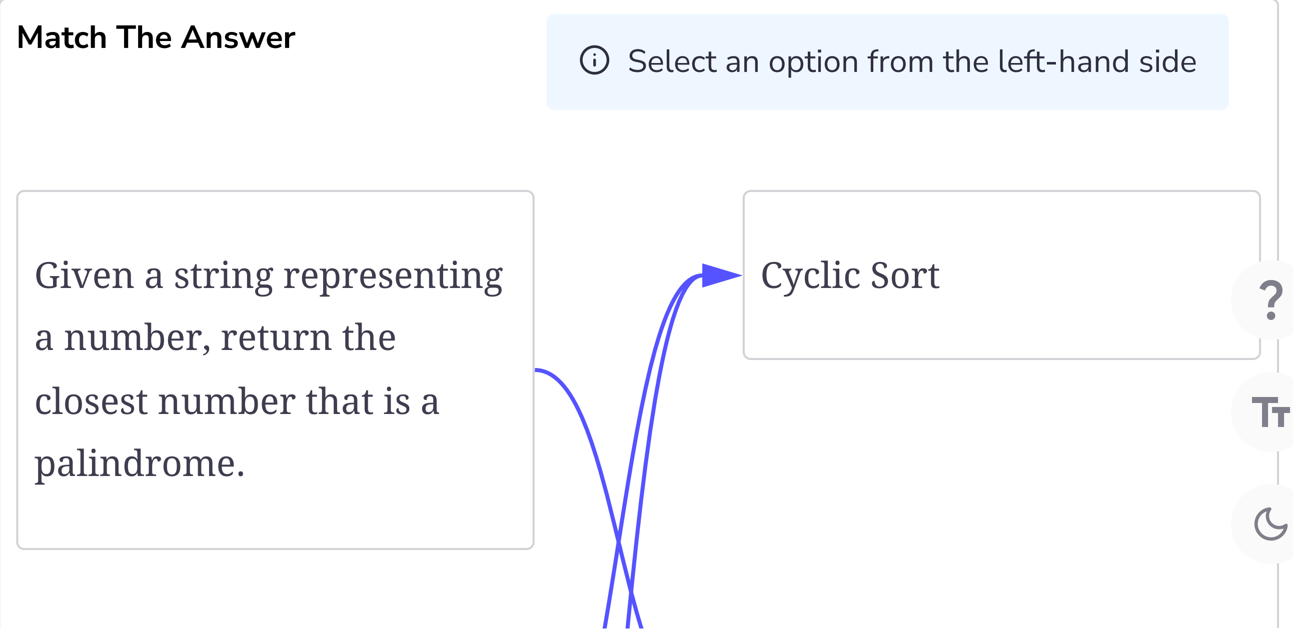
Many problems in the real world share the cyclic sort pattern. Let's look at some examples.

- **Computational Biology:** The species on a planet have n distinct genes numbered $1 \ldots n$. Find the k^{th} missing gene in a given DNA sequence.
- **Error checking:** Network transmission errors cause a number in a known range to be misinterpreted as a different number from the same range. Find the number introduced by the error, along with the number that it replaced.

Strategy time!

Match the problems that can be solved using the cyclic sort pattern.

> **Note:** Select a problem in the left-hand column by clicking it, and then click one of the two options in the right-hand column.

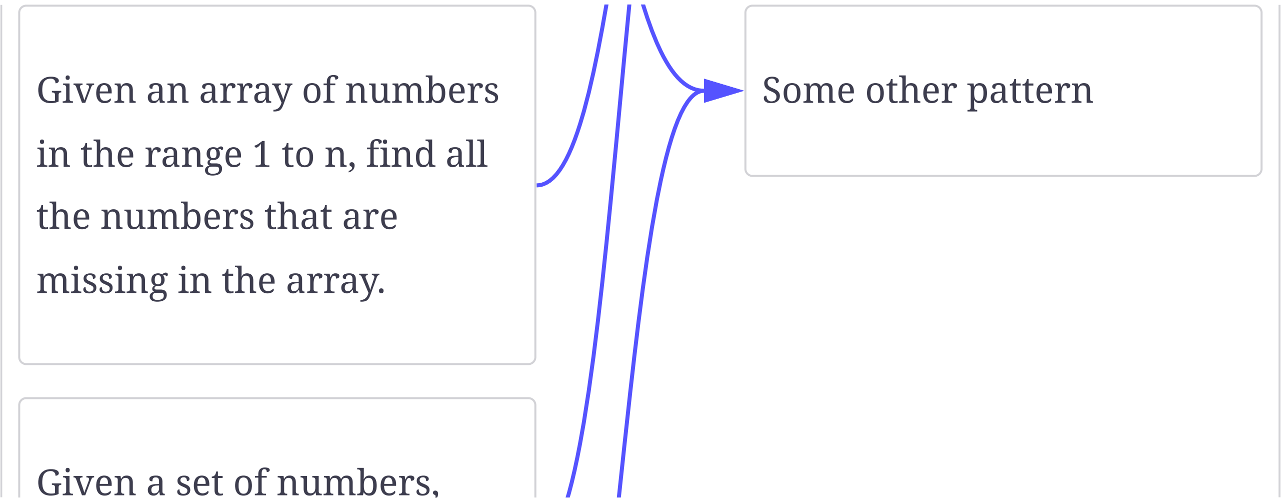


Match The Answer

ⓘ Select an option from the left-hand side

Given a string representing a number, return the closest number that is a palindrome.

Cyclic Sort



Given an array of numbers in the range 1 to n, find all the numbers that are missing in the array.

Some other pattern

Given a set of numbers,