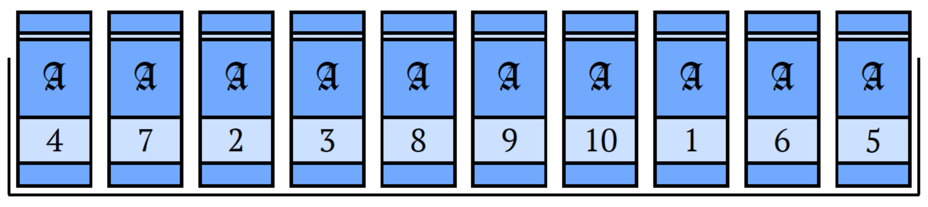
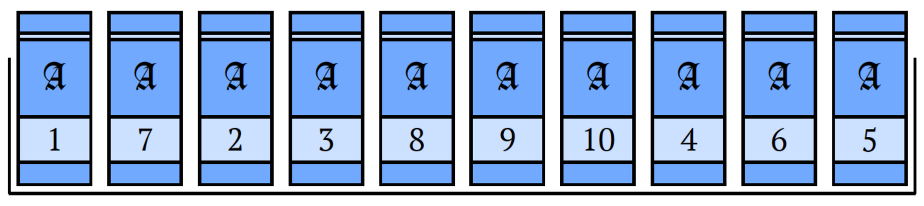
Arthur’s Books

In the following problem, we will use invariants twice — to show that something can be done efficiently and to show that there also exists a limit to how efficiently it can be done.

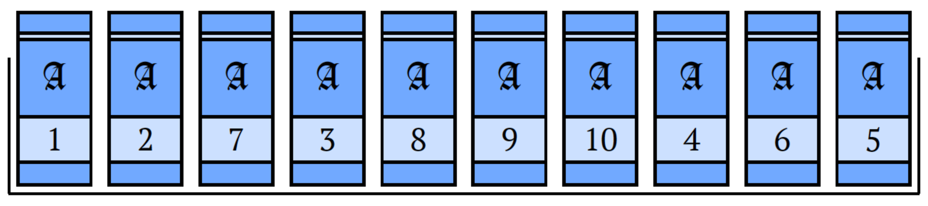
Problem. King Arthur has a shelf of his works consisting of 10 volumes, numbered $1, 2, \dots, 10$. Over the years of use, the volumes got disordered. Arthur hires Merlin to sort the collection, but he doesn't want more than two volumes to leave the shelf at once. The volumes are heavy, so Merlin can only switch two volumes on the shelf in a day. How many days does Merlin need to surely sort the volumes regardless of their initial position?



Let us show that nine days are enough for any initial permutation of the volumes. On the first day, Merlin finds the position j of volume 1 (in the example above, $j = 8$). If $j \neq 1$, he exchanges the volumes at the positions 1 and j.



On the second day, he finds the position j of volume 2. Note that j cannot be equal to 1, since the first position is occupied by volume 1. If $j \neq 2$, Merlin exchanges the books at positions 2 and j.



Proceeding in the same fashion, on day i, Merlin ensures that volume i moves to position i. This way, he maintains the following *invariant*: after i days, the first i volumes occupy their intended positions. It remains to note that by the end of the ninth day, volume 10 must be at position 10: volumes $1, 2, \dots, 9$ occupy positions $1, 2, \dots, 9$, hence 10 is the only available position for volume 10.

It is particularly easy to implement this strategy in Python. The code below uses 0-based indexing for days, books, and positions.

```
def sort_books(books):
    day = 0

    for i in range(len(books)):
        j = books.index(i)
        if j != i:
            books[i], books[j] = books[j], books[i]
            print(f'After day {day}: {books}')
            day += 1


sort_books([0, 5, 8, 1, 2, 3, 7, 4, 9, 6])
```

Run

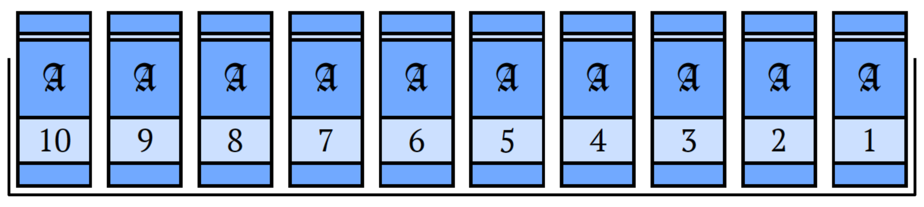
Reset

```
After day 0: [0, 1, 8, 5, 2, 3, 7, 4, 9, 6]
After day 1: [0, 1, 2, 5, 8, 3, 7, 4, 9, 6]
After day 2: [0, 1, 2, 3, 8, 5, 7, 4, 9, 6]
After day 3: [0, 1, 2, 3, 4, 5, 7, 8, 9, 6]
After day 4: [0, 1, 2, 3, 4, 5, 6, 8, 9, 7]
After day 5: [0, 1, 2, 3, 4, 5, 6, 7, 9, 8]
After day 6: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
After day 0: [0, 1, 8, 5, 2, 3, 7, 4, 9, 6]
After day 1: [0, 1, 2, 5, 8, 3, 7, 4, 9, 6]
After day 2: [0, 1, 2, 3, 8, 5, 7, 4, 9, 6]
After day 3: [0, 1, 2, 3, 4, 5, 7, 8, 9, 6]
After day 4: [0, 1, 2, 3, 4, 5, 6, 8, 9, 7]
After day 5: [0, 1, 2, 3, 4, 5, 6, 7, 9, 8]
After day 6: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
None
```

This example reveals the following property: at least some permutations (orderings) can be sorted in fewer than nine days.

Are there permutations of ten books that cannot be sorted in fewer than nine days?

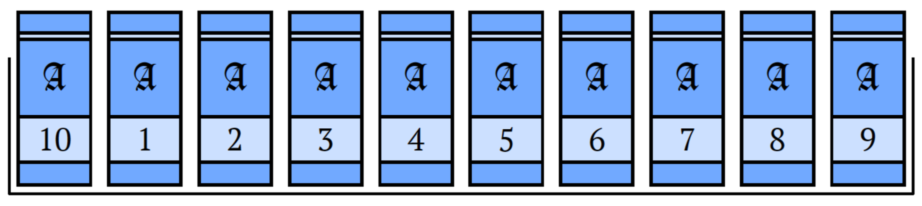
A natural first guess would be to consider the volumes sorted in the reverse order.



However, it can be sorted in just five days:

$$10 \leftrightarrow 1, \quad 9 \leftrightarrow 2, \quad 8 \leftrightarrow 3, \quad 7 \leftrightarrow 4, \quad 6 \leftrightarrow 5\,.$$

Consider the following permutation.



Try to sort it manually or to run the code above on it (don't forget to subtract one from the volume numbers, as the code assumes 0-based indexing). Note that if you (and the Python function sort_books) are not able to sort it in fewer than nine days, this *does not* mean that it is impossible. Indeed, Merlin's strategy (implemented above) is to place book i correctly on day i, for $i = 1, 2, \dots, 9$. However, there are many other strategies: say, on the first day, we may want to place book 7 correctly, then book 2, and so on. What if some of these strategies allow for sorting the above sequence in fewer than nine days? Below, we exploit invariants to prove formally that no such strategy exists.

Consider the following integer measure of a permutation π: $\mathrm{inv}(\pi)$ is the number of elements located to the right of their intended positions. The following example highlights such elements.

$$\begin{aligned}
&\mathrm{inv}([1,2,3,4,5,6,7,8,9,10]) = 0\,,\\
&\mathrm{inv}([10,9,8,7,6,5,4,3,2,1]) = 5\,,\\
&\mathrm{inv}([3,1,4,8,6,10,5,9,2,7]) = 4\,,\\
&\mathrm{inv}([10,1,2,3,4,5,6,7,8,9]) = 9\,.
\end{aligned}$$

Roughly speaking, all highlighted elements should move to the leftand all the remaining elements should either move to the right or stay unchanged. Since each exchange moves one element to the left and one element to the right, the number of days required should be at least the number of highlighted elements. We make this formal below.

We claim that a permutation π cannot be sorted in fewer than $\mathrm{inv}(\pi)$ days. Indeed, when π is sorted, $\mathrm{inv}(\pi) = 0$ (when all volumes occupy their intended places, no volume stands to the right of its intended position). Each day, one book moves rightward and one book moves leftward. Only the latter book may contribute to reducing the value of $\mathrm{inv}(\pi)$, so the value of $\mathrm{inv}(\pi)$ drops by at most one. All of this allows us to conclude that

$$\pi = [10, 1, 2, 3, 4, 5, 6, 7, 8, 9]$$

cannot be sorted in fewer than nine days.

Completed

Go to next item