Bubble Sort

Testleaf
Always Ahead

- One of the oldest sorting algorithm
- Run time Complexity : $O(n^2)$

Example

Sort the given array

11	4	17	18	2	22	1	8

Compare Adjacent

11	4	17	18	2	22	1	8

Step 1) Compare the adjacent value

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

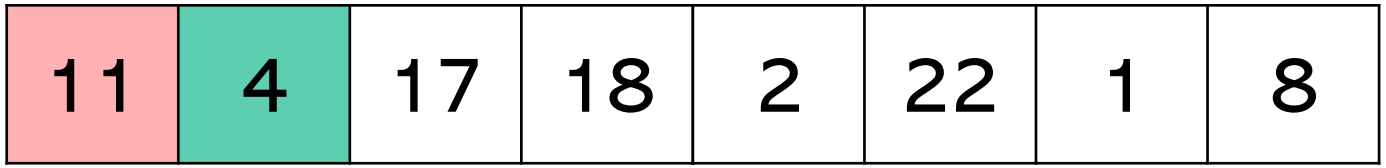
Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare

11	4	17	18	2	22	1	8

Step 1) Compare the index 0 and 1

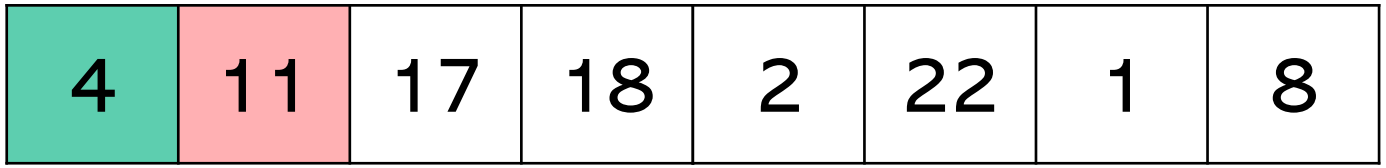
Compare



Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

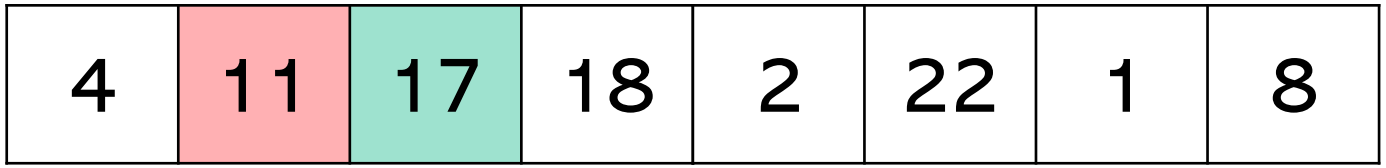
Swap



Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Compare, Swap and Continue

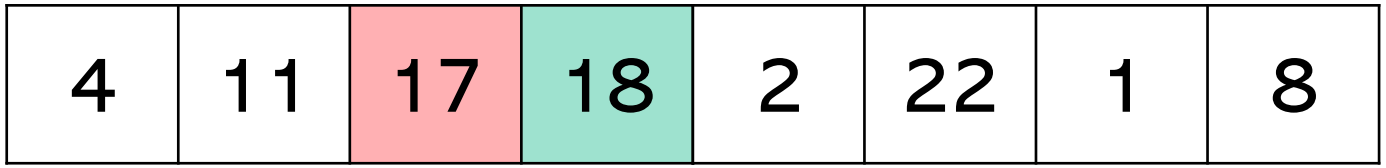


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Compare, Swap and Continue

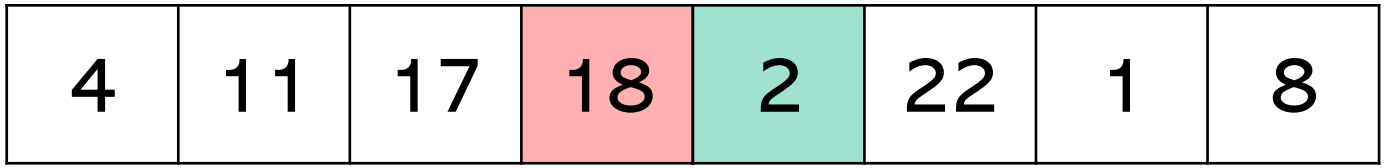


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Compare, Swap and Continue

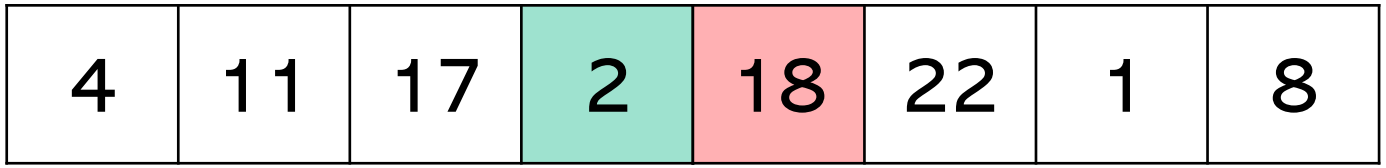


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Swap

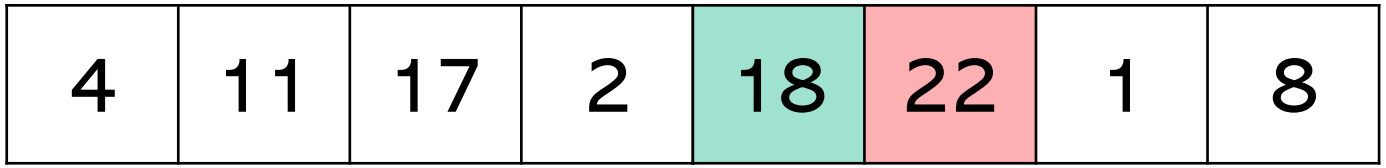


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Compare, Swap and Continue

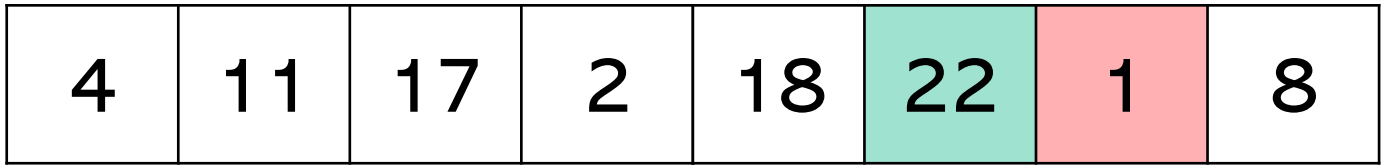


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Compare, Swap and Continue

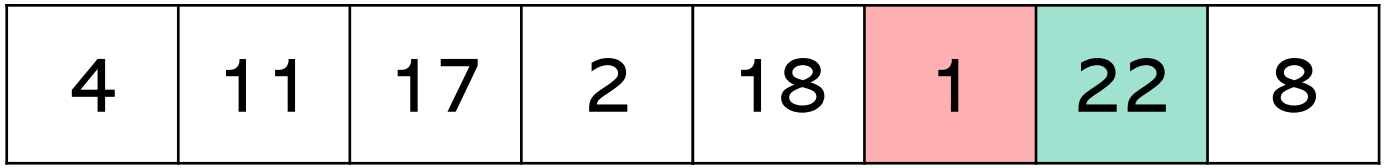


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Swap

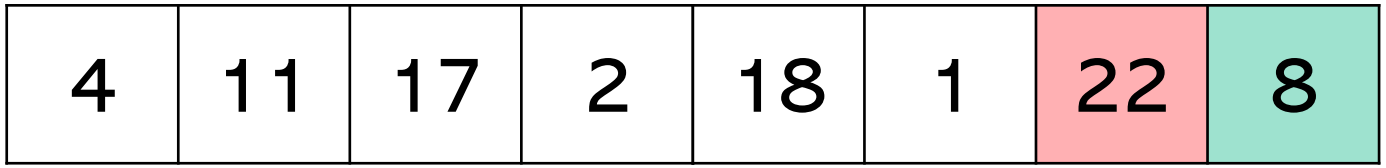


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Compare, Swap and Continue

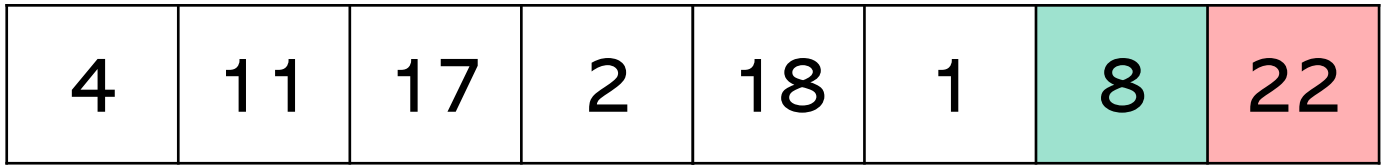


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Swap

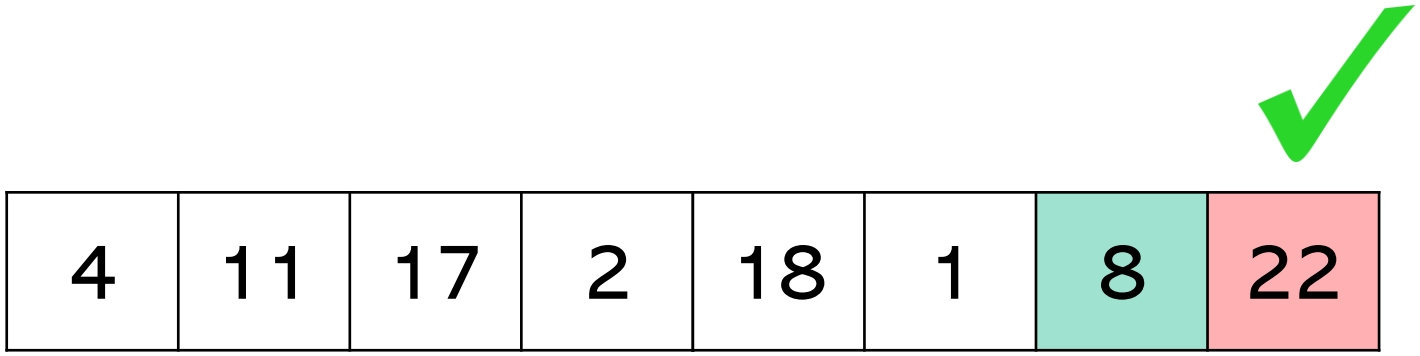


Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Next Iteration



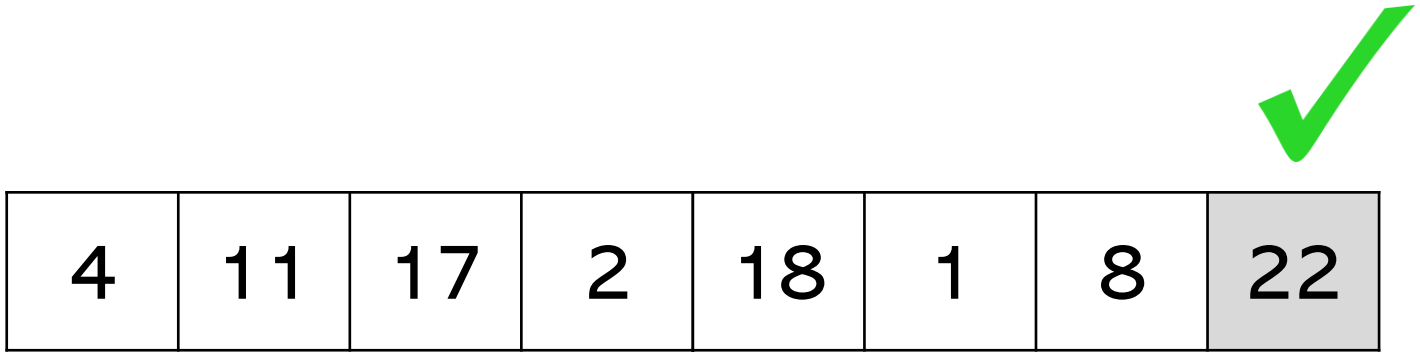
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Next Iteration



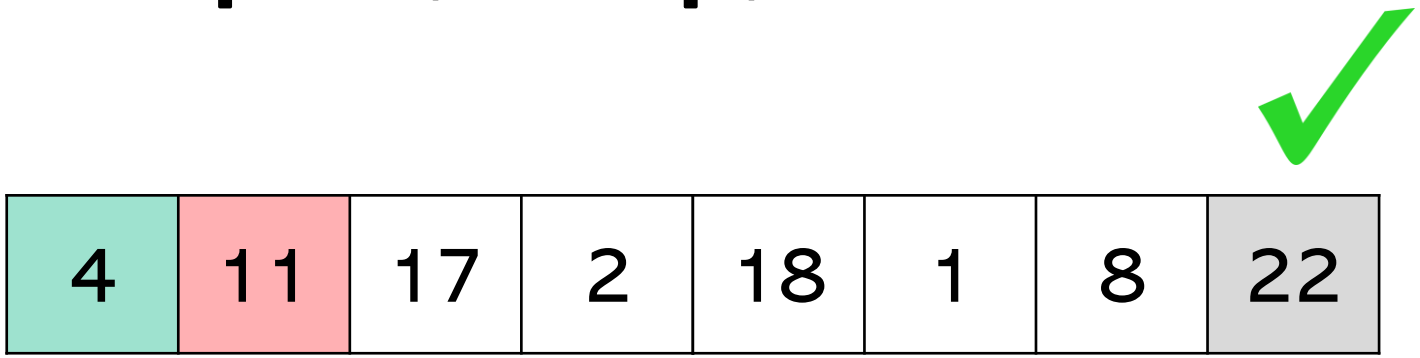
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



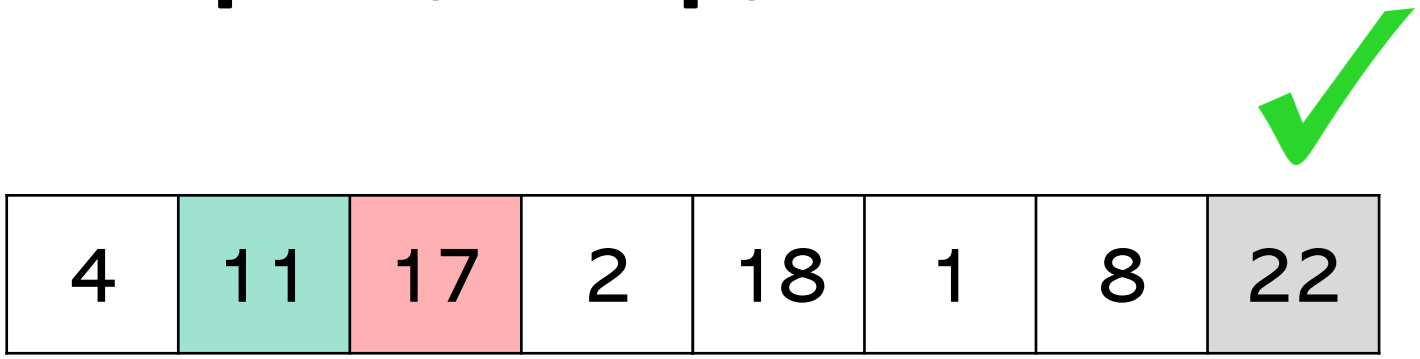
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



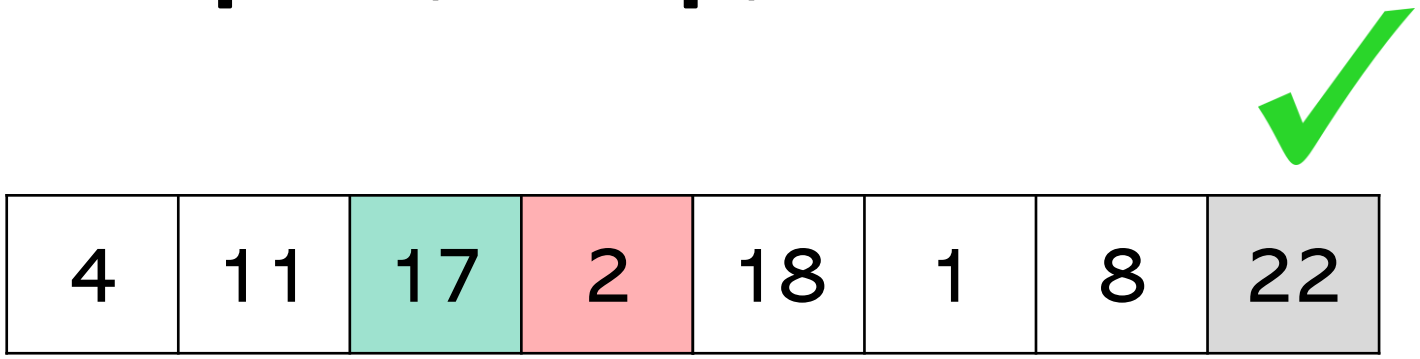
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



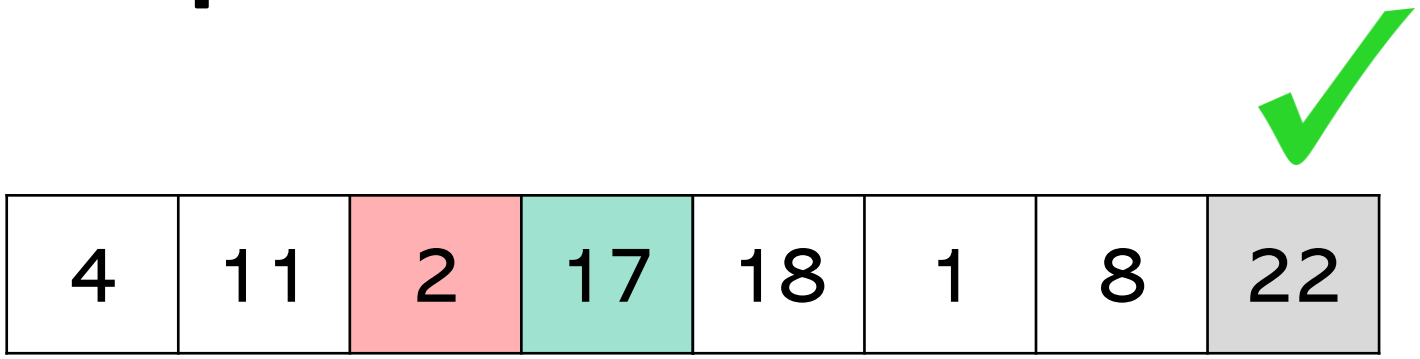
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Swap



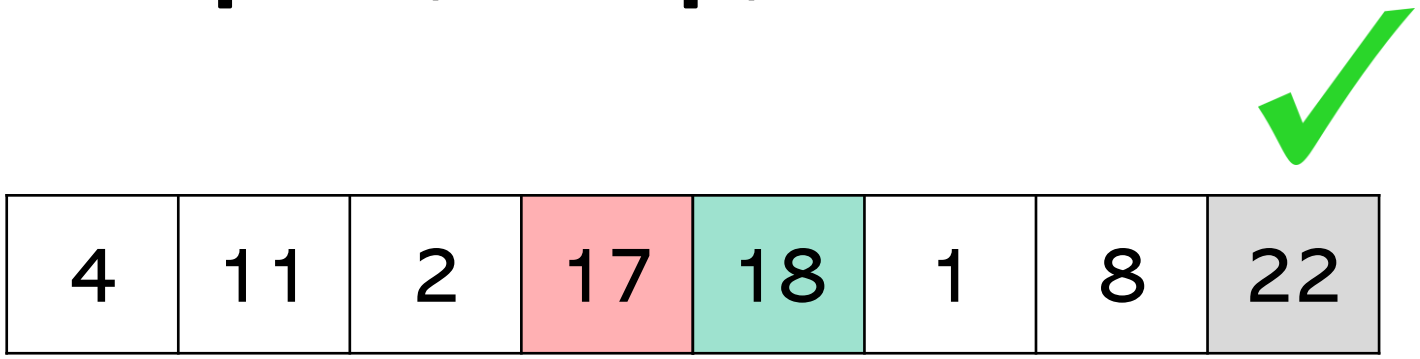
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



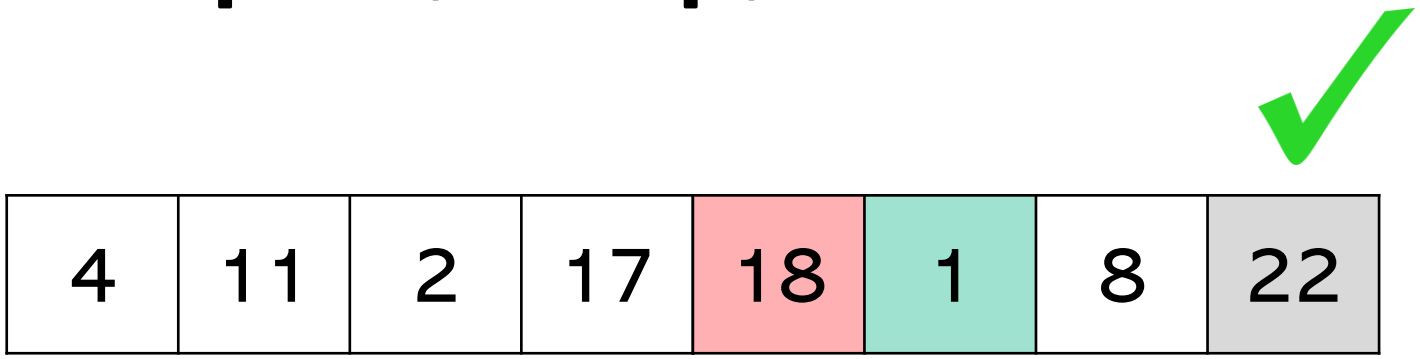
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



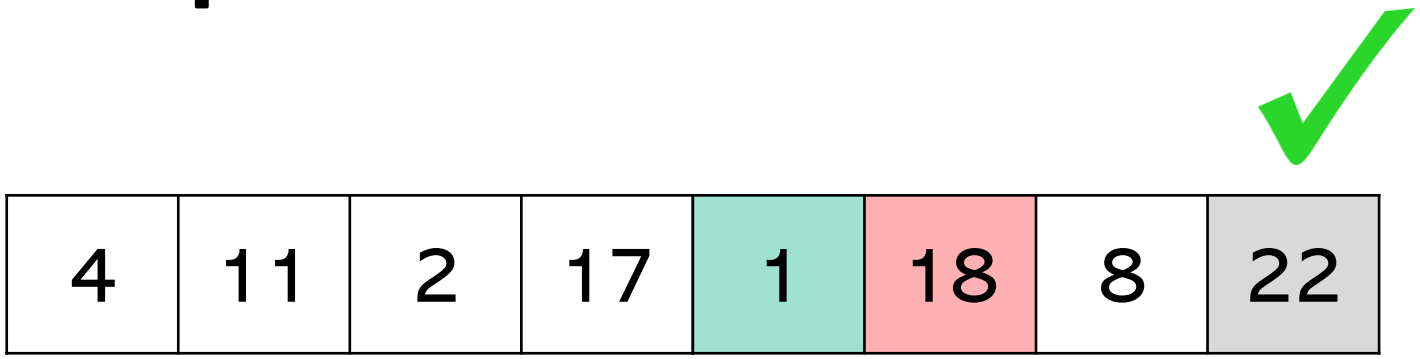
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Swap



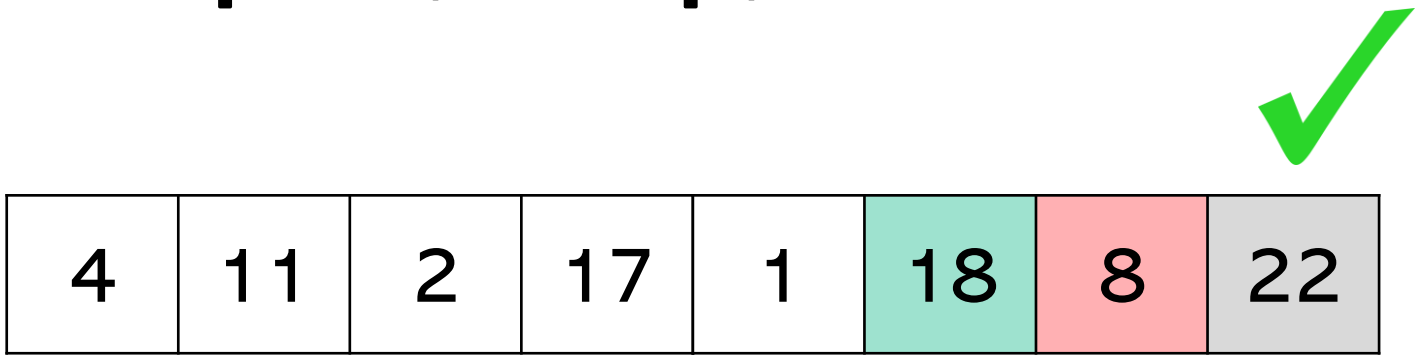
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Compare, Swap, Continue



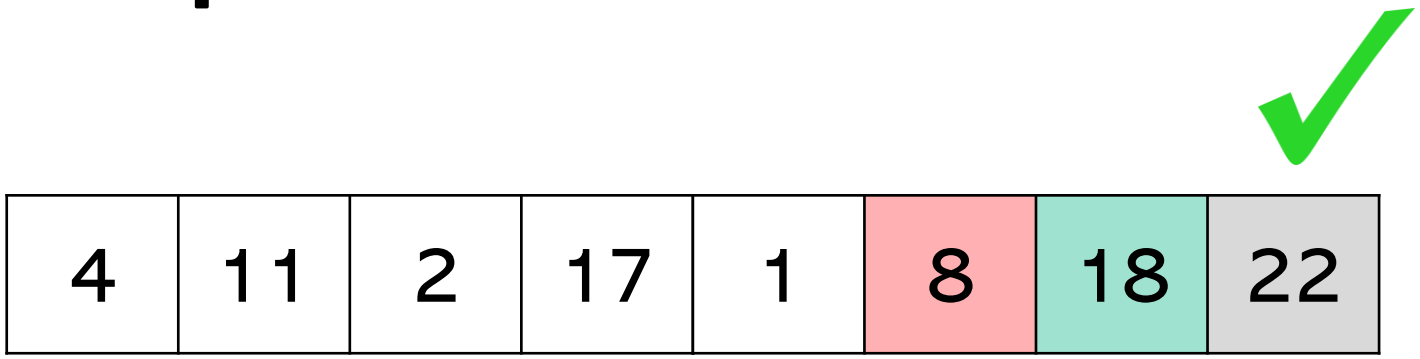
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Swap



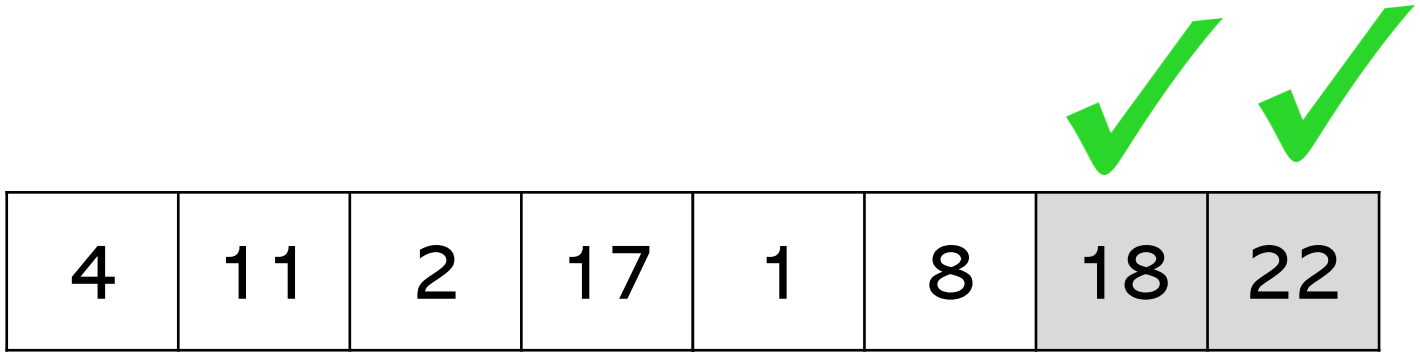
Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)

Next Iteration



4	11	2	17	1	8	18	22

Step 1) Compare the index 0 and 1

Step 2) If the left index is greater than the right index, then swap

Step 3) Continue Step 2 from 0 to n

Step 4) Continue Step 2 & 3 from 1 to (n-i)