Insertion Sort

Siri Chandana Mandadapu

10/10/2022

CS-601-01

Agenda

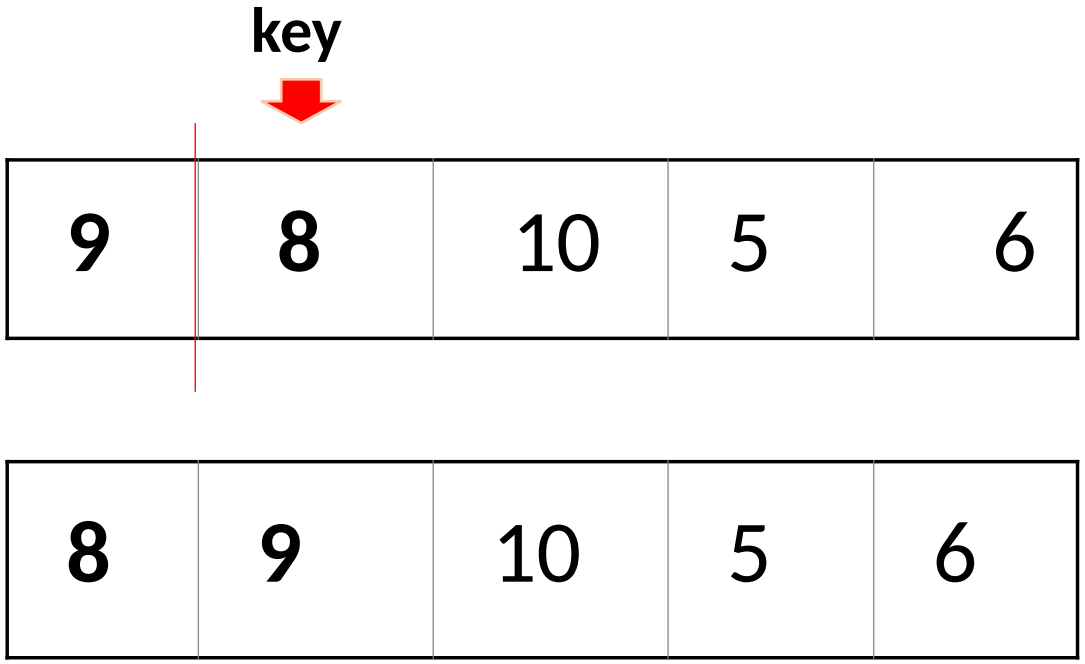
- Insertion sort definition
- Working and Algorithm
- Practice problem
- Pseudo Code
- Time complexity
- Advantages
- Disadvantages

What is Insertion sort?

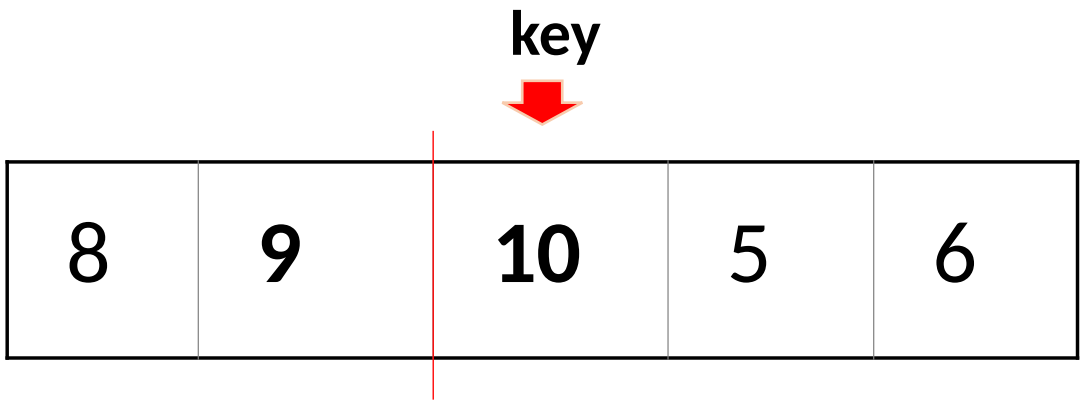
- Simple sorting algorithm
- It sorts an arbitrary array, one element at a time by incrementing the index of an array and compare the element with the previous elements(to the left) to find the suitable spot.
- Used in computer programs such as file search, data compression, and path finding.

Working of Insertion sort

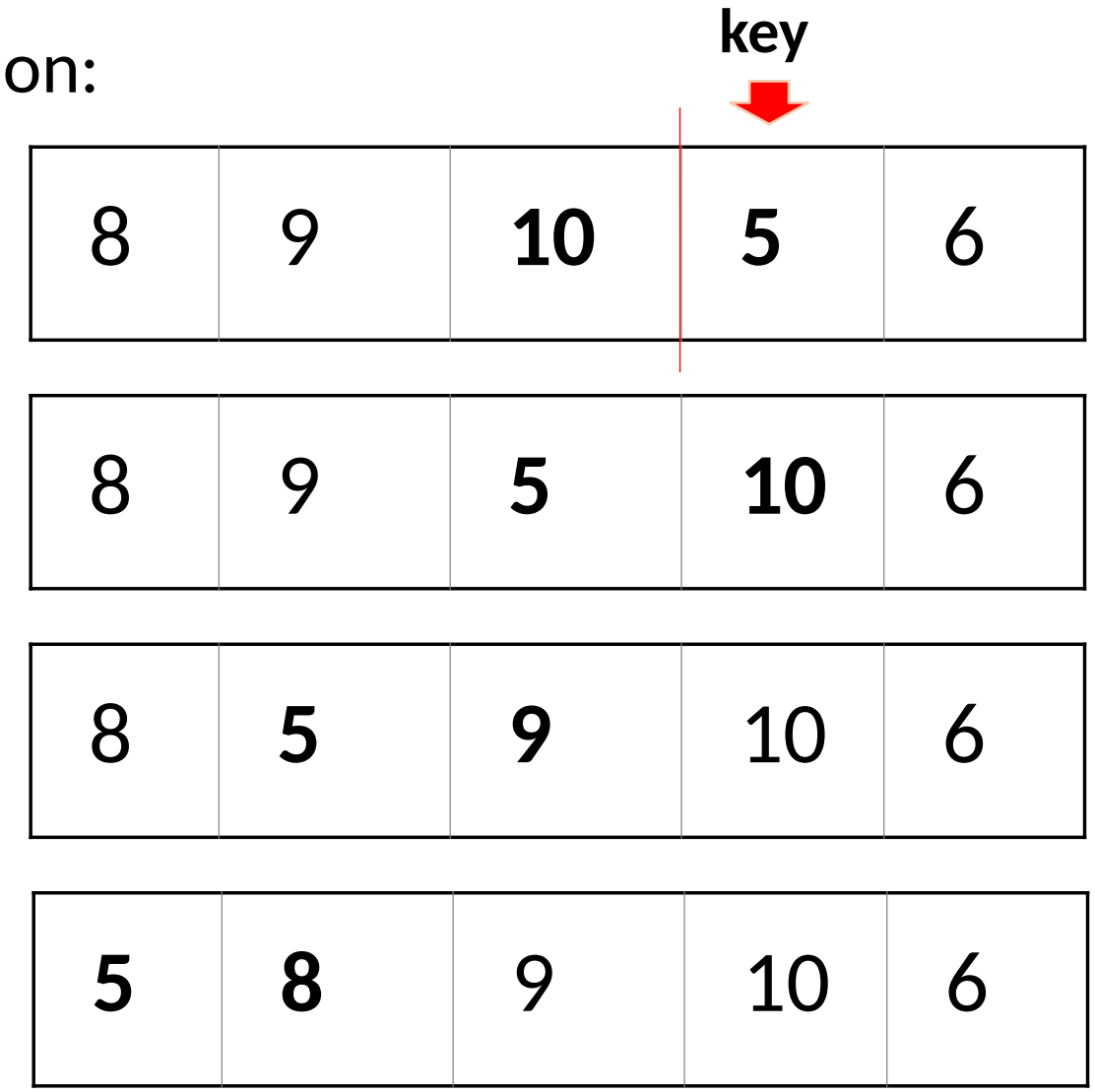
First Iteration:



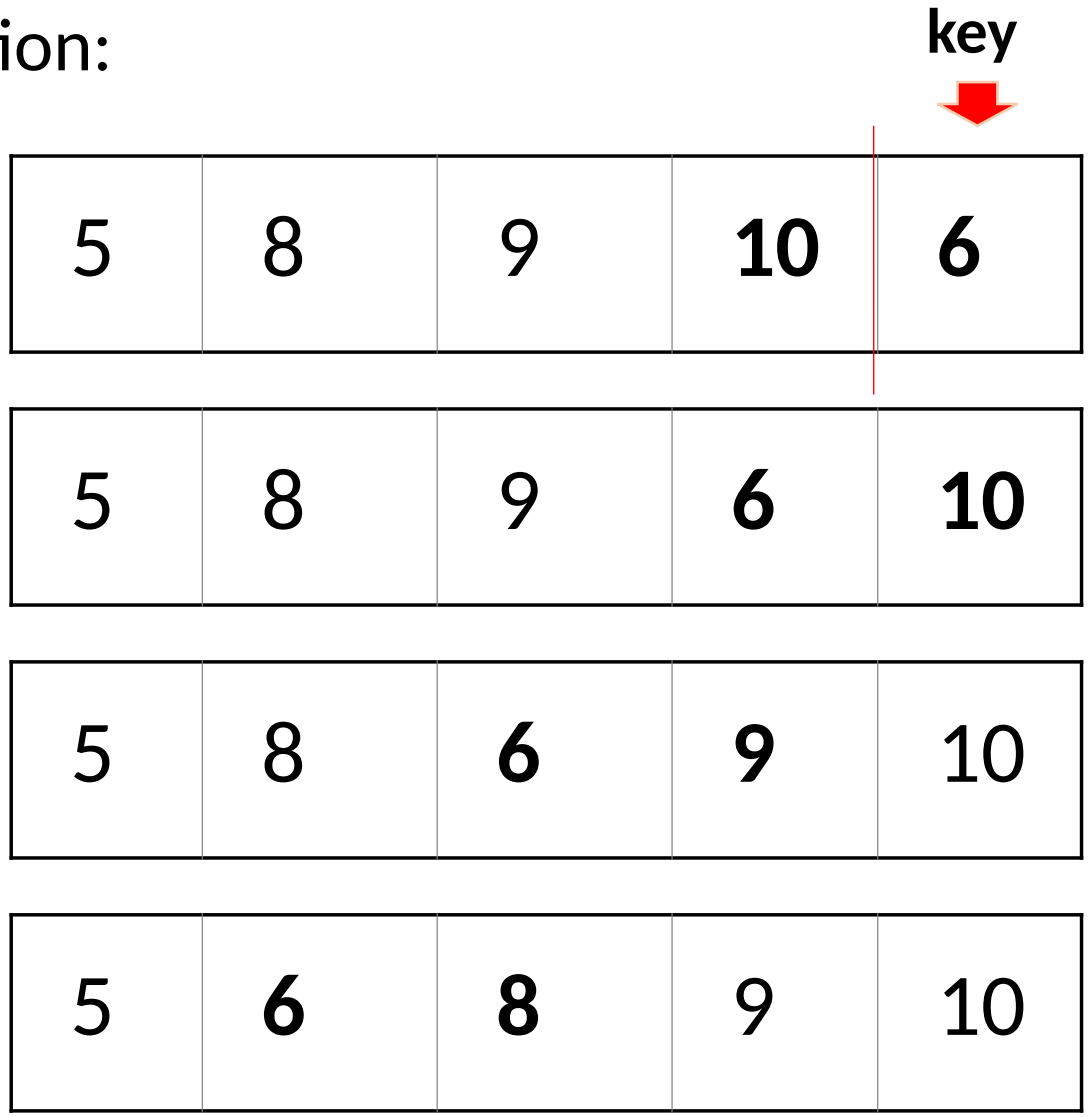
Second Iteration:



Third Iteration:



Fourth Iteration:



Practice problem

Array: 6,2,9,3,1,5,8,7,4

Solution

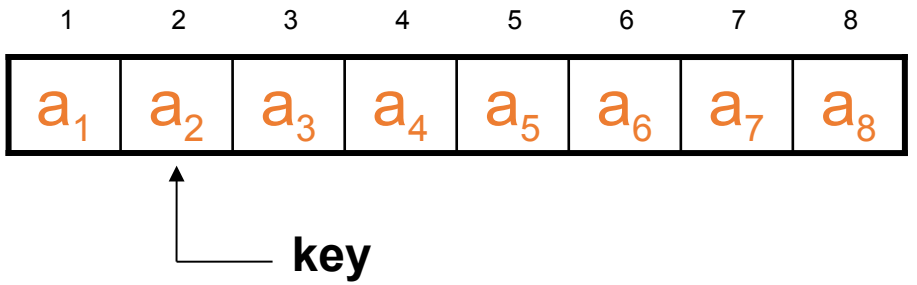
Array: 6,2,9,3,1,5,8,7,4

Final Sorted array: 1,2,3,4,5,6,7,8,9

Pseudo Code

```
void insertionSort(int arr[], int n)
{
   int i, key, j;
   for (i = 1; i < n; i++)
   {
      key = arr[i];
      j = i - 1;

      while (j >= 0 && arr[j] > key)          //comparing the elements
      {
         arr[j + 1] = arr[j];
         j = j - 1;
      }
      arr[j + 1] = key;                 // swapping
   }
}
```



Time Complexity - Best Case Analysis

- The array is already sorted
  - A[i] ≤ key is true for all the comparisons. (where i=j-1)

1	2	3	4	5

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5(n-1) + c_8(n-1)$$

$$= (c_1 + c_2 + c_4 + c_5 + c_8)n + (c_2 + c_4 + c_5 + c_8)$$

$$T(n) = cn + c' = O(n)$$

Time Complexity - Worst Case Analysis

5	4	3	2	1

- Array is in reverse sorted order
  - $A[i] > key$ in **while** loop is always true till the end of array
  - Compare key with all elements to the left

- We have a nested loop each of size n in the worst-case, so the time complexity will be of polynomial degree 2

$$T(n) = O(n^2)$$

Time complexity summary

- For Best case scenario - O(n)
- For Worst Case or the Average Case Scenario – $O(n^2)$

THANK YOU!