OUTPUT

Average Case:
When few elements are ordered:

int a[6]={5,8,9,6,7,10};

After pass 1:
5 8 9 6 7 10
5 8 9 6 7 10
5 8 6 9 7 10
5 8 6 7 9 10
5 8 6 7 9 10
After pass 2:
5 8 6 7 9 10
5 6 8 7 9 10
5 6 7 8 9 10
5 6 7 8 9 10
After pass 3:
5 6 7 8 9 10
5 6 7 8 9 10
5 6 7 8 9 10

The sorted array is:
5 6 7 8 9 10

Time Complexity:
$O(n^2)$

Worst Case:
When none of the elements are sorted:

int a[6]={10,9,8,7,6,5};

After pass 1:
9 10 8 7 6 5
9 8 10 7 6 5
9 8 7 10 6 5
9 8 7 6 10 5
9 8 7 6 5 10
After pass 2:
8 9 7 6 5 10
8 7 9 6 5 10
8 7 6 9 5 10
8 7 6 5 9 10
After pass 3:
7 8 6 5 9 10
7 6 8 5 9 10
7 6 5 8 9 10
After pass 4:
6 7 5 8 9 10
6 5 7 8 9 10
After pass 5:
5 6 7 8 9 10

The sorted array is:
5 6 7 8 9 10

Best Case:
When all the elements are sorted:

int a[6]={5,6,7,8,9,10};

After pass 1:
5 6 7 8 9 10
5 6 7 8 9 10
5 6 7 8 9 10
5 6 7 8 9 10
5 6 7 8 9 10

The sorted array is:
5 6 7 8 9 10

Time Complexity:
$O(n)$

CONCLUSION: Hence, in this modified algorithm we use a flag variable, that indicates the status of array. We do not need to again sort a sorted array(Best Case), thereby reducing large number of iterations.