PSEUDOCODE :-

1. finding the maximum element from the end of the array and storing the occurring maximums in a separate array
2. then setting an outer loop from 0 to n-1
3. and applying binary search on the maximums array we created.
4. maximising the answer each time we get a desired value.

```
int maxm[n+5];
// filling maxm array with INT_MIN

for(int i=n-1; i>=0; i–)
{
maxm[i] = max(max[i+1] , a[i]);
}

int maxDist = INT_MIN;
for(int i=0; i<n; i++)
{
  int low = 0, high = n-1 ;
  int ans = i;
  while(low<=high)
{
 int mid = (low+high)/2;
 if(a[i] <= maxm[mid])
 {
 ans = max(ans, mid);
 low = mid+1;
}
else
{
high = mid-1;
}
}
 maxDist = max(maxDist , ans - i);
}

if(maxDist == INT_MIN)
return -1;
```

```
else
return maxDist;
```

Time Complexity :- O(nlogn)
Space Complexity :- O(n)