|  |  |
| --- | --- |
| **Sorting Algorithm** | **Presoude Code** |
| Insertions Sort | For a =2 to n  Key=A[2]  b=a-1  while b > A[b] > Key  A[b+1]=A[b]  b—  A[b+1]=key |
| Selection Sort | for a =1 to A.length  min=a  for j =a+1 to A.length  if A [j] < A [min]:  min=j  if min!= a :  Swap A[a],A[min] |
| Merge Sort | MergeSort (A ,low ,high )  if low < high:  m= ((low+high-1)/2)  MergeSort(A , low ,mid)  MergeSort(A ,mid+1 ,high)  Merge(A, low ,mid ,high)  Merge ( A , low , mid , high)  Create two array left and right and store the data  compare left and right element which one is small add in a new array after this done store data should be given to A. |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sorting Algorithm** | **Best Case**  **Time Complexity** | **Worst Case**  **Time Complexity** | **Average Case**  **Time Complexity** | **Strength of the sorting algorithm** | **Weeknesses** |
| Insertions Sort | O(n) | O(n^2) | O(n^2) | **1.** Stable algorithm means first element of duplicate come first the second duplicate element.  **2.** Perform well in small data set.  **3.** Space require is minimal. | **1.** Does not deal well for the huge list  **2.** Does not perform well as other algorithm  **3.** Useful for only small list |
| Selection Sort | O(n^2) | O(n^2) | O(n^2) | **1.** Arrange of data does not matter  **2.** Perform well in small data set  **3.** | **1.** Does not deal well for the huge list  **2.** Scanning of the whole array  **3.** unstable algorithm |
| Merge Sort | O(nlogn) | O(nlogn) | O(nlogn) | **1.**Stable algorithm means first element of duplicate come first the second duplicate element  **2.** Perform well for larger data set  **3.** | **1.** Goes through whole process if the list is sorted  **2.** Usage of memory  **3.** Slow for small data set |
| Bubble Sort | O(n) | O(n^2) | O(n^2) |  |  |
| Bucket | O(n+k) | O(n^2) | O(n+k) |  |  |
| Quick Sort | O(nlogn) | O(n^2) | O(nlogn) |  |  |
| Counting Sort | O(n+k) | O(n+k) | O(n+k) |  |  |