Date _____

Expt. No. _____ Page No. _____

Insertion Sort

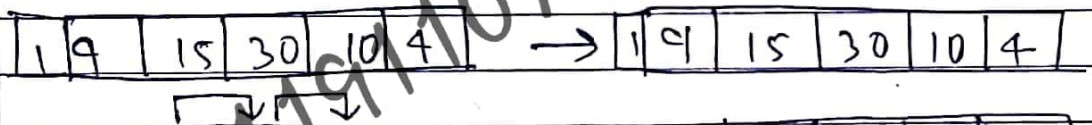
The algorithm is based on one assumption that a single element is always sorted.

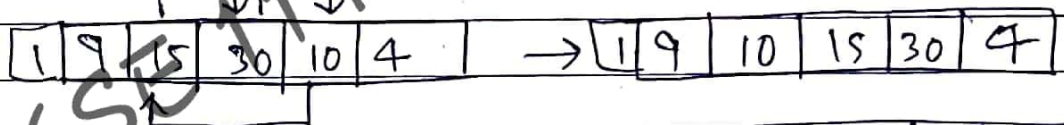
Working principle

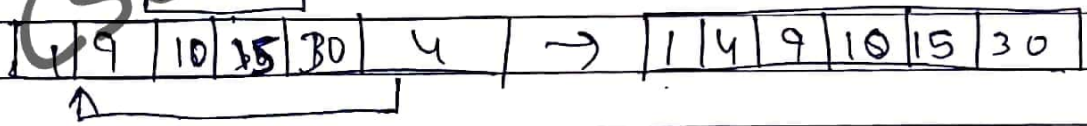
* compare the element with adjacent element

Example

Input → 1, 15, 9, 30, 10, 4, 35







Complexity

Worst case - $\Theta(n^2)$

Average case - $\Theta(n^2)$

Best case - $\Theta(n)$

for the best case array is already sorted hence only one time we have to check.

Teacher's Signature _____

Optimization

Searching & swapping are two main function

① Optimising searching by using BST which reduce the search complexity from $O(n)$ to $O(\log n)$ for one element and for n element $O(n \log n)$

② we can optimize the swapping by using doubly linked list. which will ~~reqr~~ reduce complexity of swapping n element from $O(n)$ to $O(n)$ as we can insert an element by changing pointers but searching remains $O(n^2)$ as we cannot use binary search in link list. Thus over all complexity remain $O(n^2)$