Experiment-3
Insertion sort

Implement Insertion Sort algorithm and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n

Decrease-and-Conquer

1. Reduce problem instance to smaller instance of the same problem
2. Solve smaller instance
3. Extend solution of smaller instance to obtain solution to original instance

- Can be implemented either top-down or bottom-up
- Also referred to as *inductive* or *incremental* approach

3 Types of Decrease and Conquer

- *Decrease by a constant* (usually by 1):
  - insertion sort
  - graph traversal algorithms (DFS and BFS)
  - topological sorting
  - algorithms for generating permutations, subsets
- *Decrease by a constant factor* (usually by half)
  - binary search and bisection method
  - exponentiation by squaring
  - multiplication à la russe
- *Variable-size decrease*
  - Euclid's algorithm
  - selection by partition
  - Nim-like games

This usually results in a recursive algorithm.

Insertion Sort

To sort array A[0..*n*-1], sort A[0..*n*-2] recursively and then insert A[*n*-1] in its proper place among the sorted A[0..*n*-2]

- Usually implemented bottom up (nonrecursively)

Example: Sort 6, 4, 1, 8, 5

$6 \mid \underline{4} \;\; 1 \;\; 8 \;\; 5$

$4 \;\; 6 \mid \underline{1} \;\; 8 \;\; 5$

$1 \;\; 4 \;\; 6 \mid \underline{8} \;\; 5$

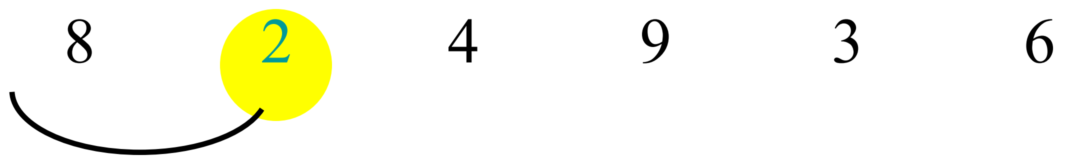
$1 \;\; 4 \;\; 6 \;\; 8 \mid \underline{5}$

$1 \;\; 4 \;\; 5 \;\; 6 \;\; 8$

Example of insertion sort

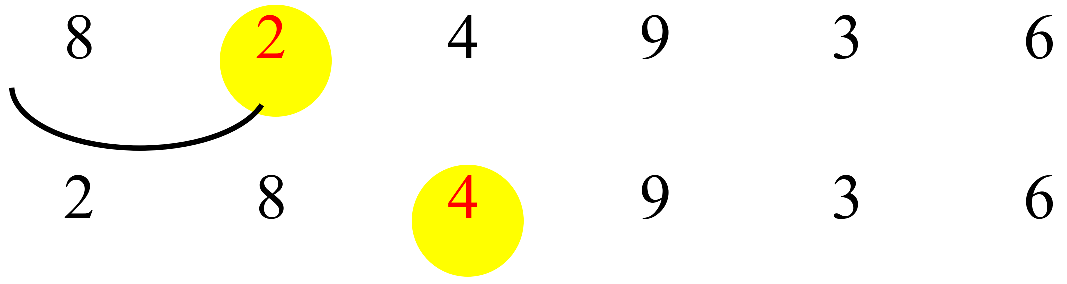
8 2 4 9 3 6

Example of insertion sort

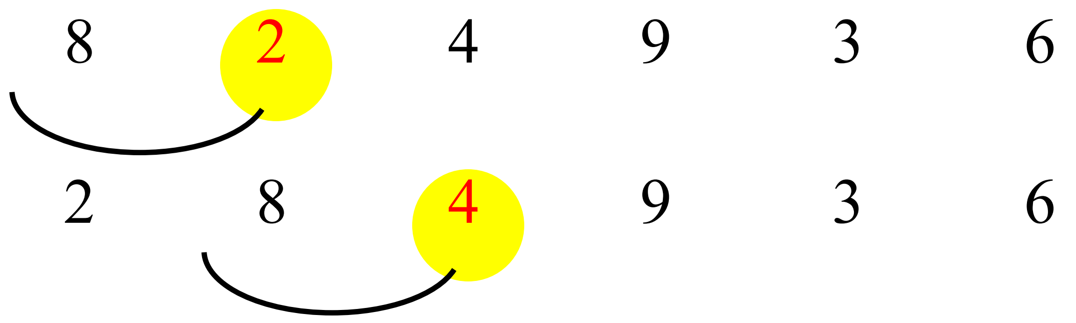


8 2 4 9 3 6

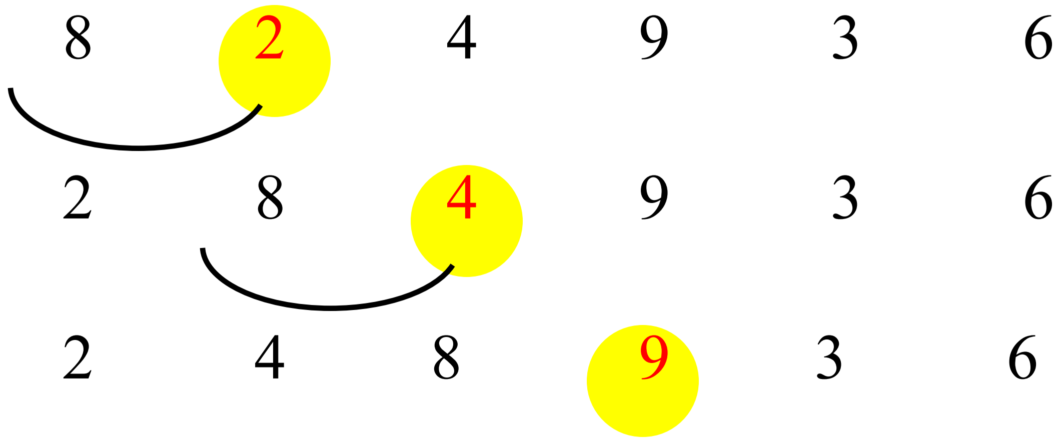
Example of insertion sort



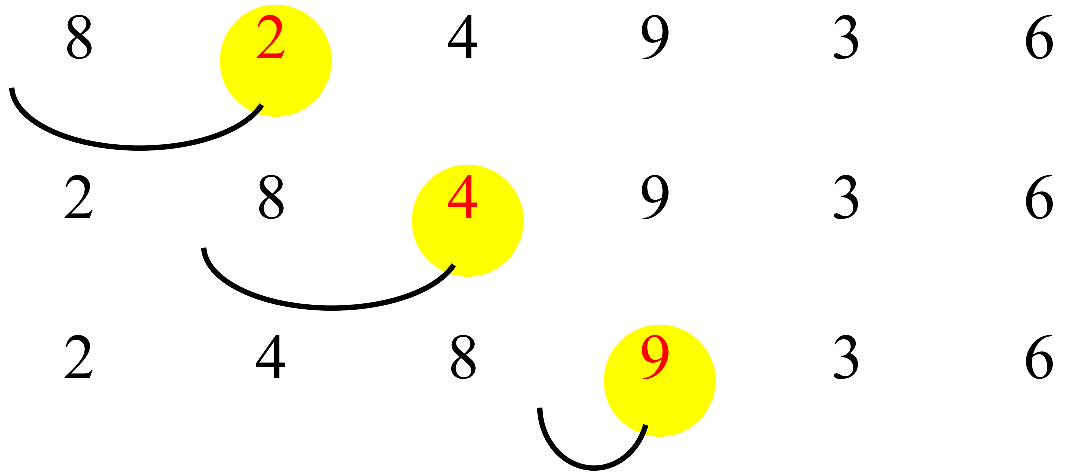
Example of insertion sort



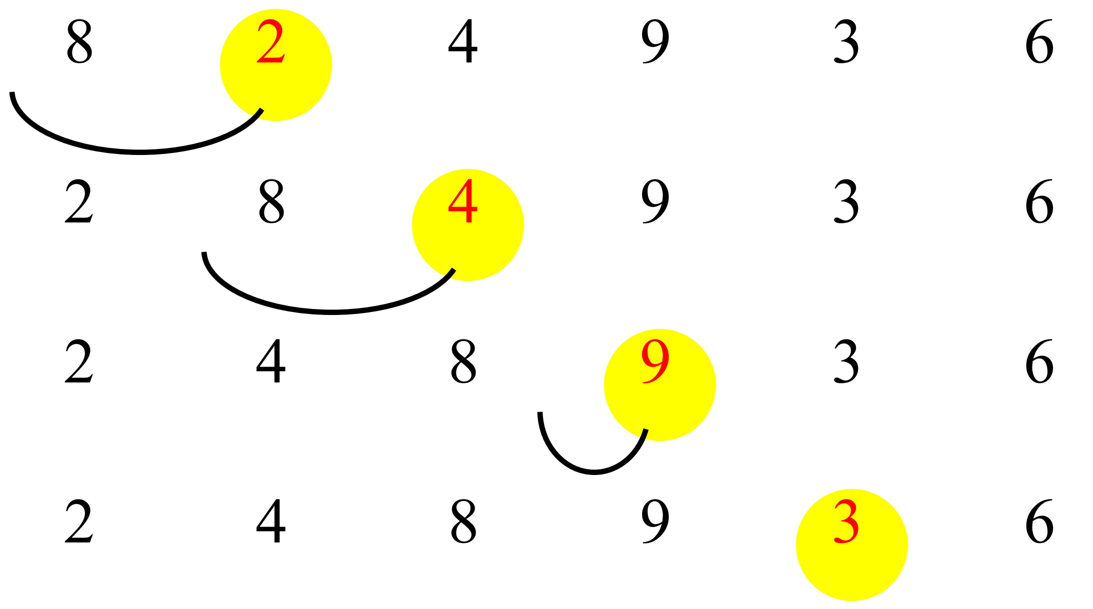
Example of insertion sort



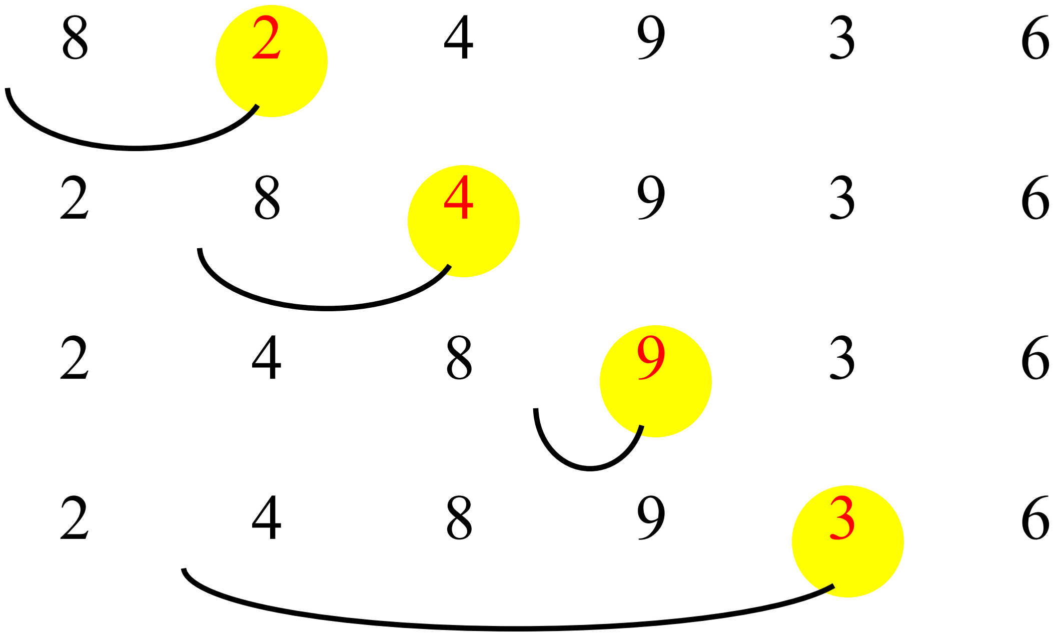
Example of insertion sort



Example of insertion sort



Example of insertion sort

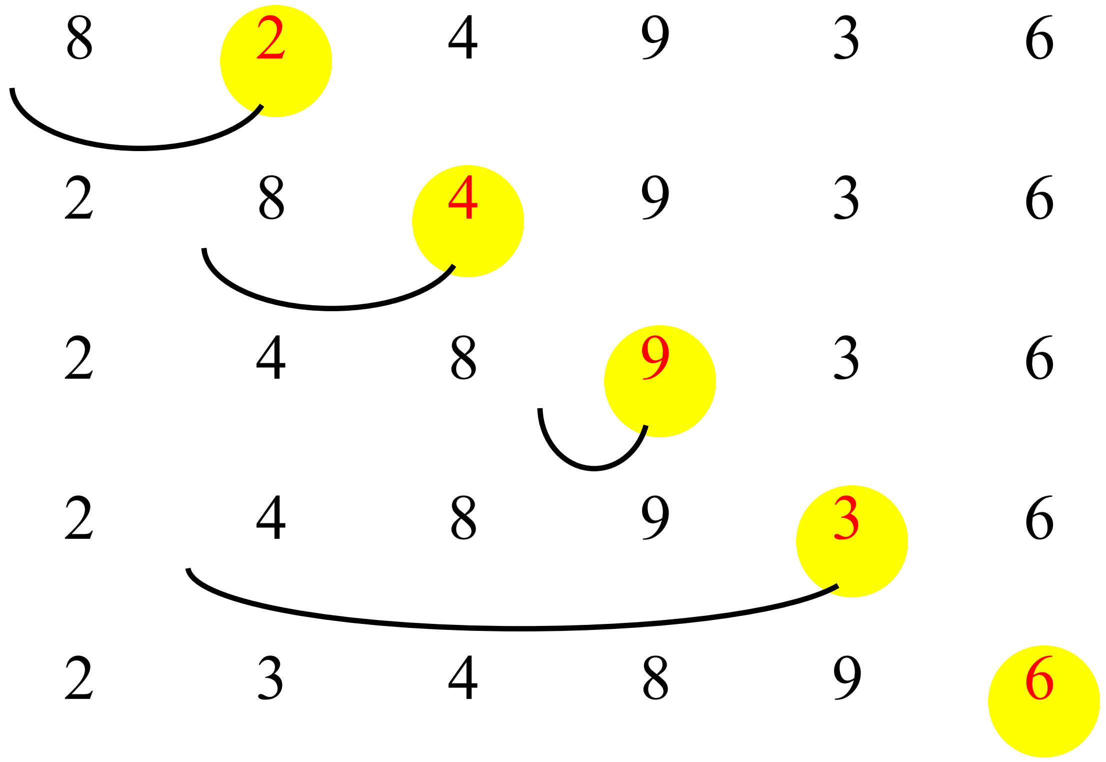

8 2 4 9 3 6

2 8 4 9 3 6

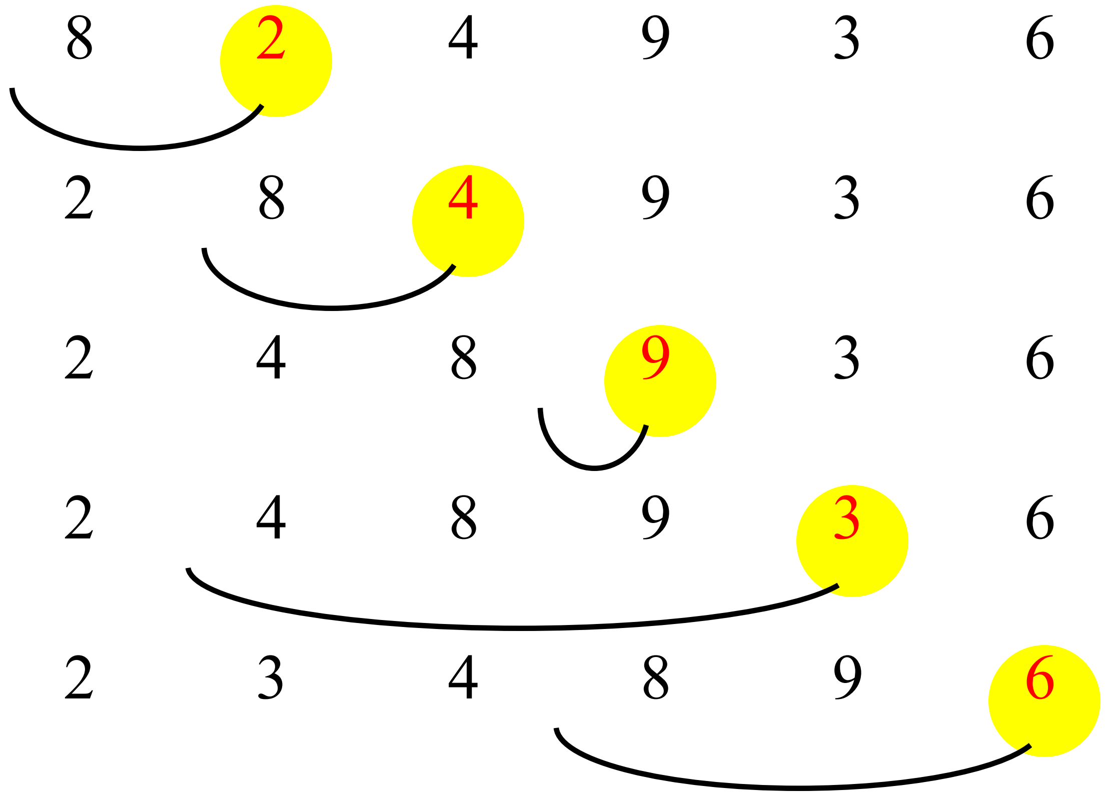
2 4 8 9 3 6

2 4 8 9 3 6

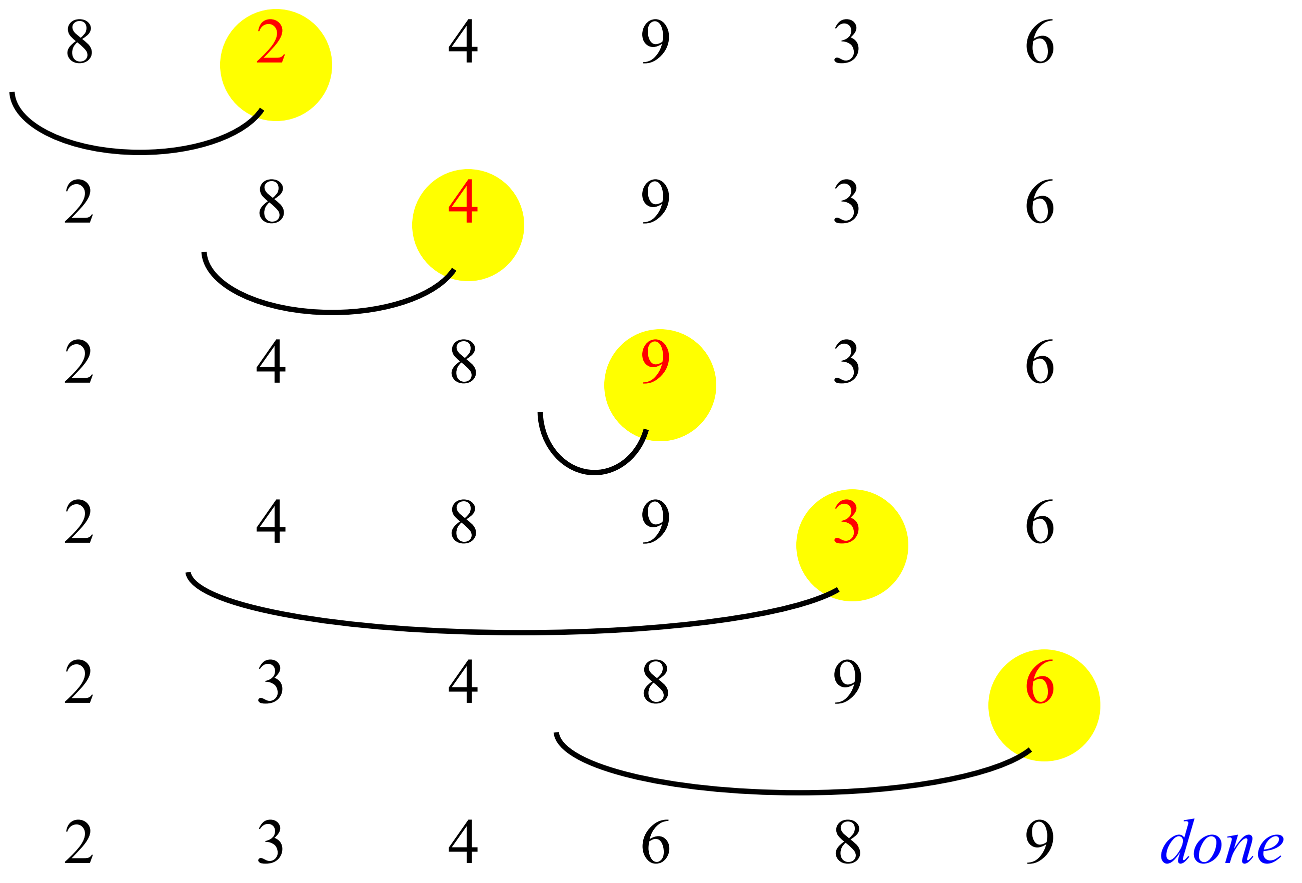
Example of insertion sort



Example of insertion sort



Example of insertion sort



Pseudocode of Insertion Sort

```
ALGORITHM  InsertionSort($A[0..n-1]$)
    //Sorts a given array by insertion sort
    //Input: An array $A[0..n-1]$ of $n$ orderable elements
    //Output: Array $A[0..n-1]$ sorted in nondecreasing order
    for $i \leftarrow 1$ to $n-1$ do
        $v \leftarrow A[i]$
        $j \leftarrow i-1$
        while $j \geq 0$ and $A[j] > v$ do
            $A[j+1] \leftarrow A[j]$
            $j \leftarrow j-1$
        $A[j+1] \leftarrow v$
```