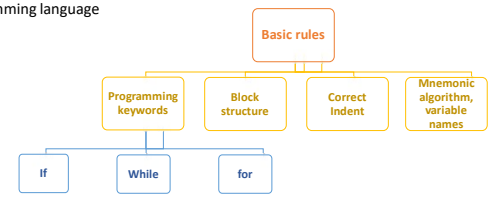
Algorithms

- Combinatorial Algorithms: Manipulates discrete objects like arrays, lists, trees, …
- Writing Algorithms:
  - Not a program, but similar to comments
  - Computational techniques for all programming languages
  - Pseudocode

1

Pseudocode

- Structure of formal programming language
- Plain English
- One-sentence steps



2

Design and Analysis of Algorithms

- ***Design***: Minimize the cost
- ***Analysis***: Predict the cost in terms of resources and performance
- Goal of an algorithm:
  - Always correct
  - Always terminate
  - Performance
    - Decide what is possible and what is not

3

Design and Analysis of Algorithms

- Big Oh notation.
- Greedy
- Divide-and-conquer.
- Dynamic programming.
- Intractability.

Emphasizes critical thinking, problem-solving, and rigorous analysis.

4

The problem of sorting

- **Input** : sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of numbers
- **Output** : permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ such that $a'_1 \le a'_2 \le \cdots \le a'_n$.
- Example :

  Input: 8 7 1 5 9 2

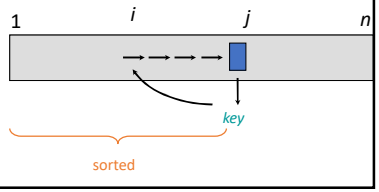
  Output: 1 2 5 7 8 9

5

Insertion sort

INSERTION-SORT (A, n)

```
1: for j ← 2 to n
2:     do key ← A[j]
3:        i ← j – 1
4:        while i > 0 and A[i] > key do
5:              A[i+1] ← A[i]
6:              i ← i – 1
7:        end while
8:        A[i+1] = key
9: end for
```



6

Example of insertion sort

8 7 1 5 9 2

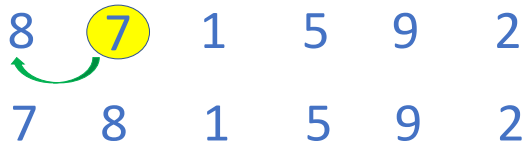
7

Example of insertion sort

8 7 1 5 9 2

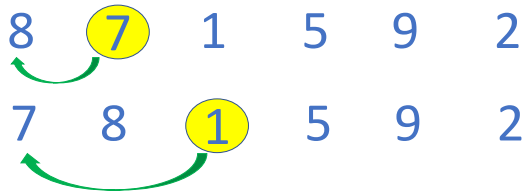
8

Example of insertion sort



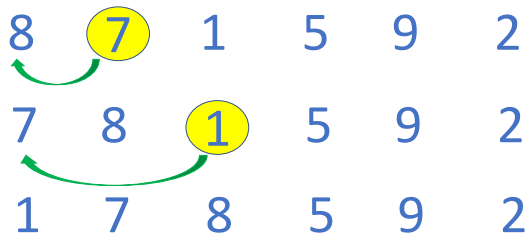
9

Example of insertion sort



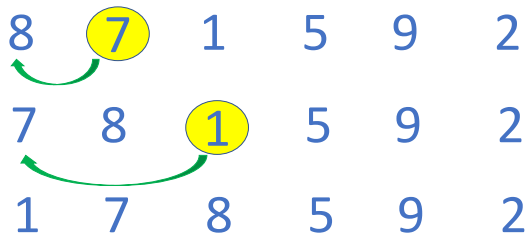
10

Example of insertion sort



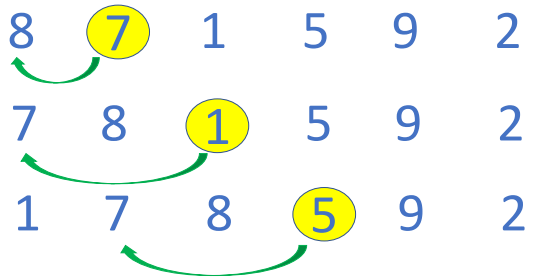
11

Example of insertion sort



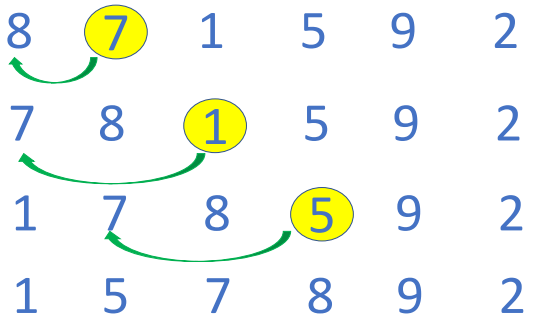
12

Example of insertion sort



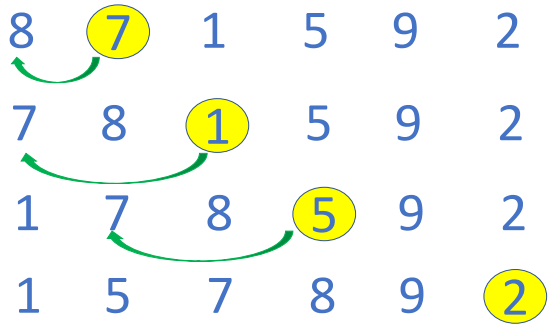
13

Example of insertion sort



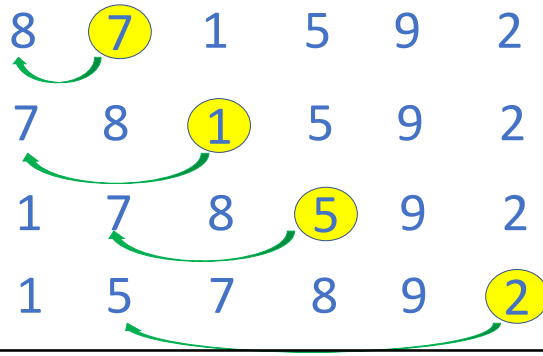
14

Example of insertion sort



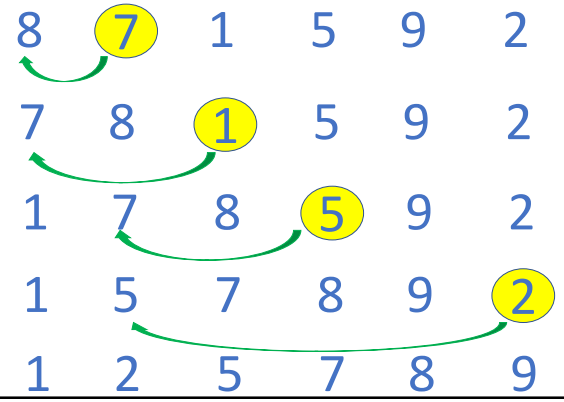
15

Example of insertion sort



16

Example of insertion sort



17

Example of insertion sort

1 2 5 7 8 9

Sorted

18