

# Insertion Sort

---

# Some Definitions

---

- Internal Sort
  - The data to be sorted is all stored in the computer's main memory.
- External Sort
  - Some of the data to be sorted might be stored in some external, slower, device.
- In Place Sort
  - The amount of extra space required to sort the data is constant with the input size.

# Stability

- A **STABLE** sort preserves relative order of records with equal keys

Sorted on first key:

Aaron	4	A	664-480-0023	097 Little
Andrews	3	A	874-088-1212	121 Whitman
Battle	4	C	991-878-4944	308 Blair
Chen	2	A	884-232-5341	11 Dickinson
Fox	1	A	243-456-9091	101 Brown
Furia	3	A	766-093-9873	22 Brown
Gazsi	4	B	665-303-0266	113 Walker
Kanaga	3	B	898-122-9643	343 Forbes
Rohde	3	A	232-343-5555	115 Holder
Quilici	1	C	343-987-5642	32 McCosh

Sort file on second key:

Fox	1	A	243-456-9091	101 Brown
Quilici	1	C	343-987-5642	32 McCosh
Chen	2	A	884-232-5341	11 Dickinson
Kanaga	3	B	898-122-9643	343 Forbes
Andrews	3	A	874-088-1212	121 Whitman
Furia	3	A	766-093-9873	22 Brown
Rohde	3	A	232-343-5555	115 Holder
Battle	4	C	991-878-4944	308 Blair
Gazsi	4	B	665-303-0266	113 Walker
Aaron	4	A	664-480-0023	097 Little

Records with key value 3 are not in order on first key!! Not sorted with a stable algorithm

# Insertion Sort

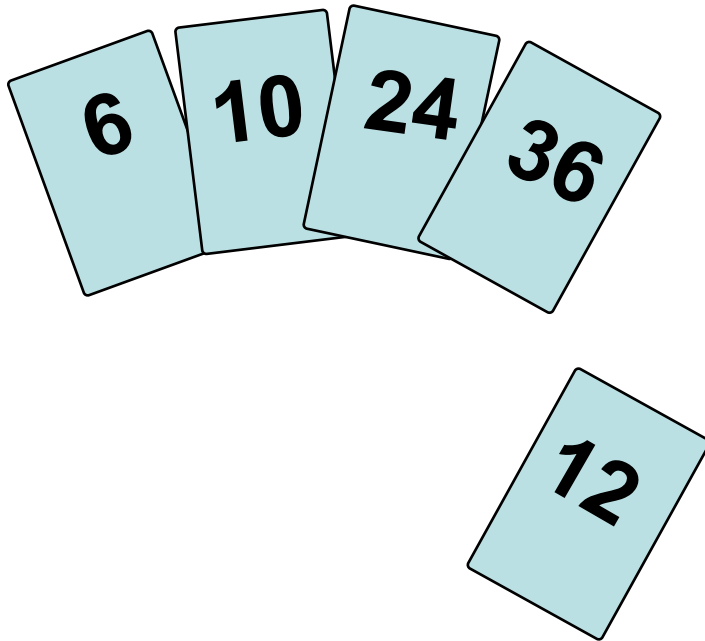
---

- Idea: like sorting a hand of playing cards
  - Start with an empty left hand and the cards facing down on the table.
  - Remove one card at a time from the table, and insert it into the correct position in the left hand
    - compare it with each of the cards already in the hand, from right to left
  - The cards held in the left hand are sorted
    - these cards were originally the top cards of the pile on the table

# Insertion Sort

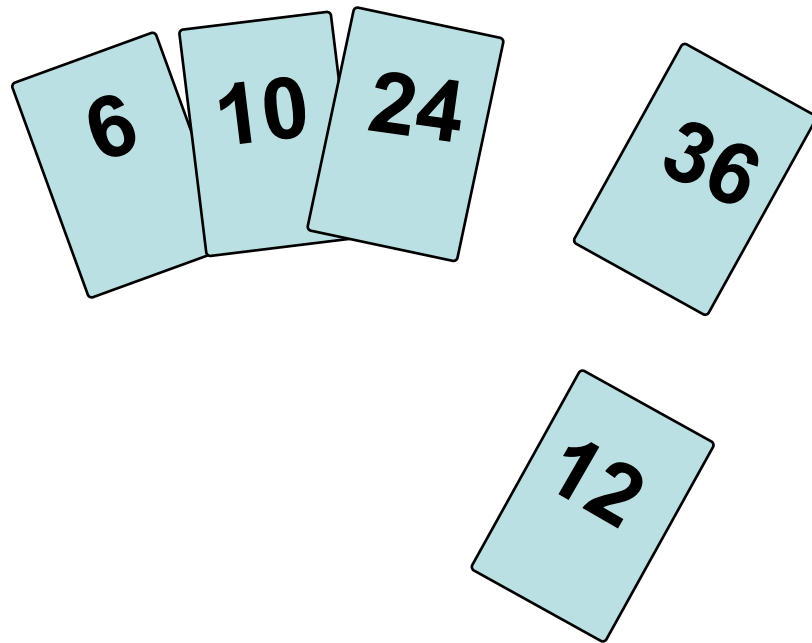
---

To insert 12, we need to make room for it by moving first 36 and then 24.



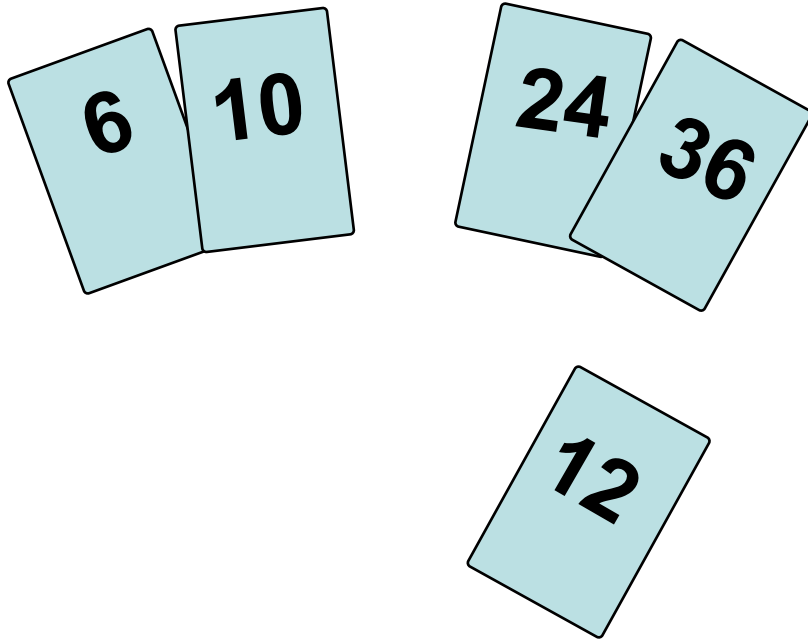
# Insertion Sort

---



# Insertion Sort

---



# Insertion Sort

---

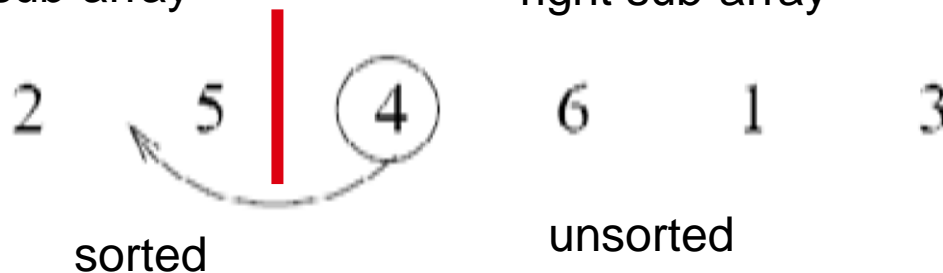
input array

5    2    4    6    1    3

at each iteration, the array is divided in two sub-arrays:

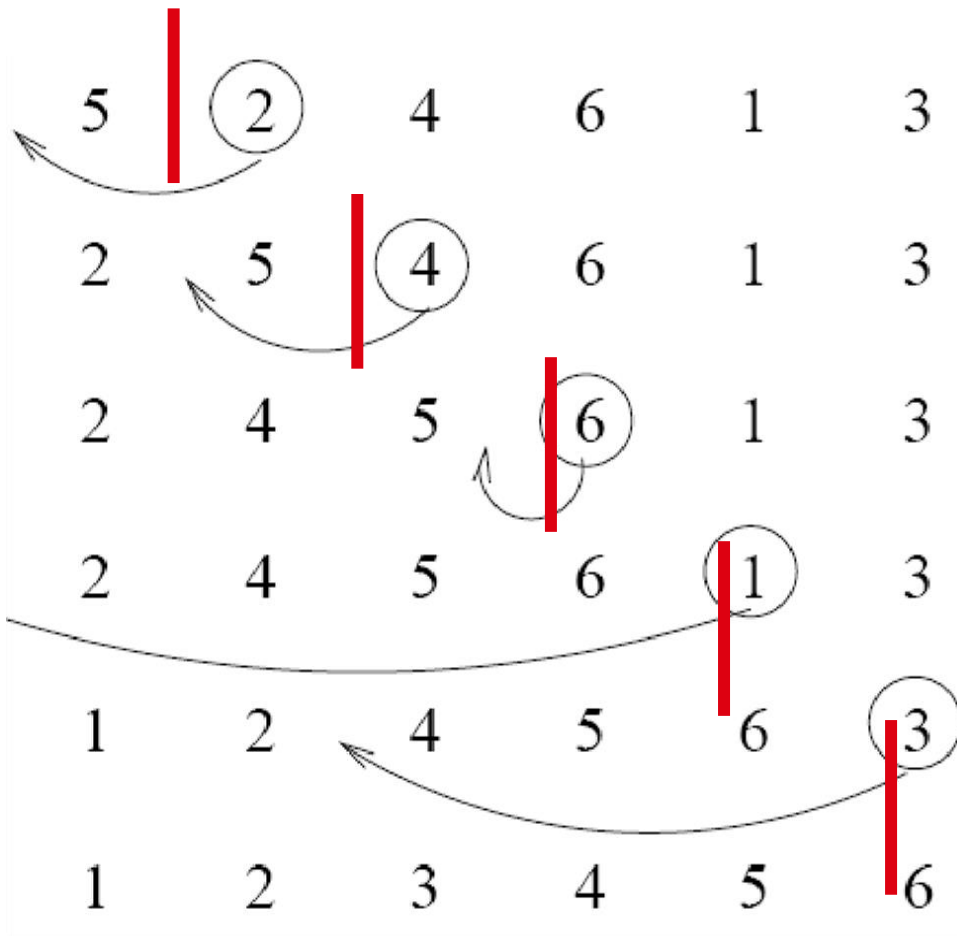
left sub-array

right sub-array

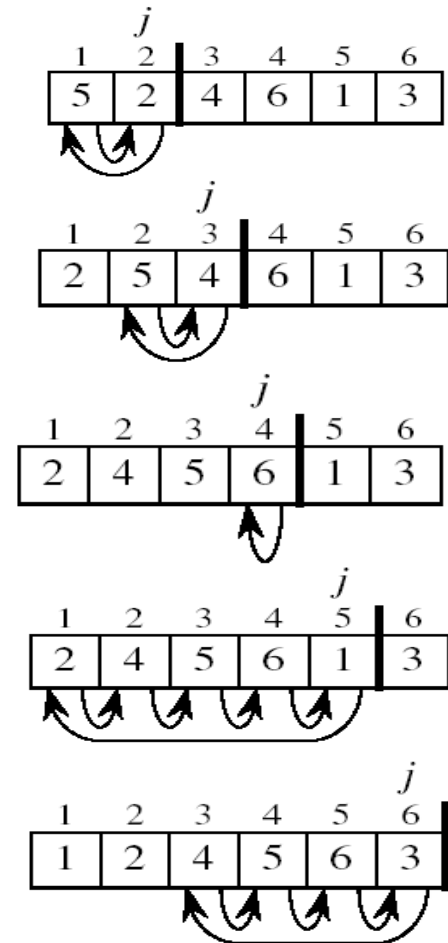
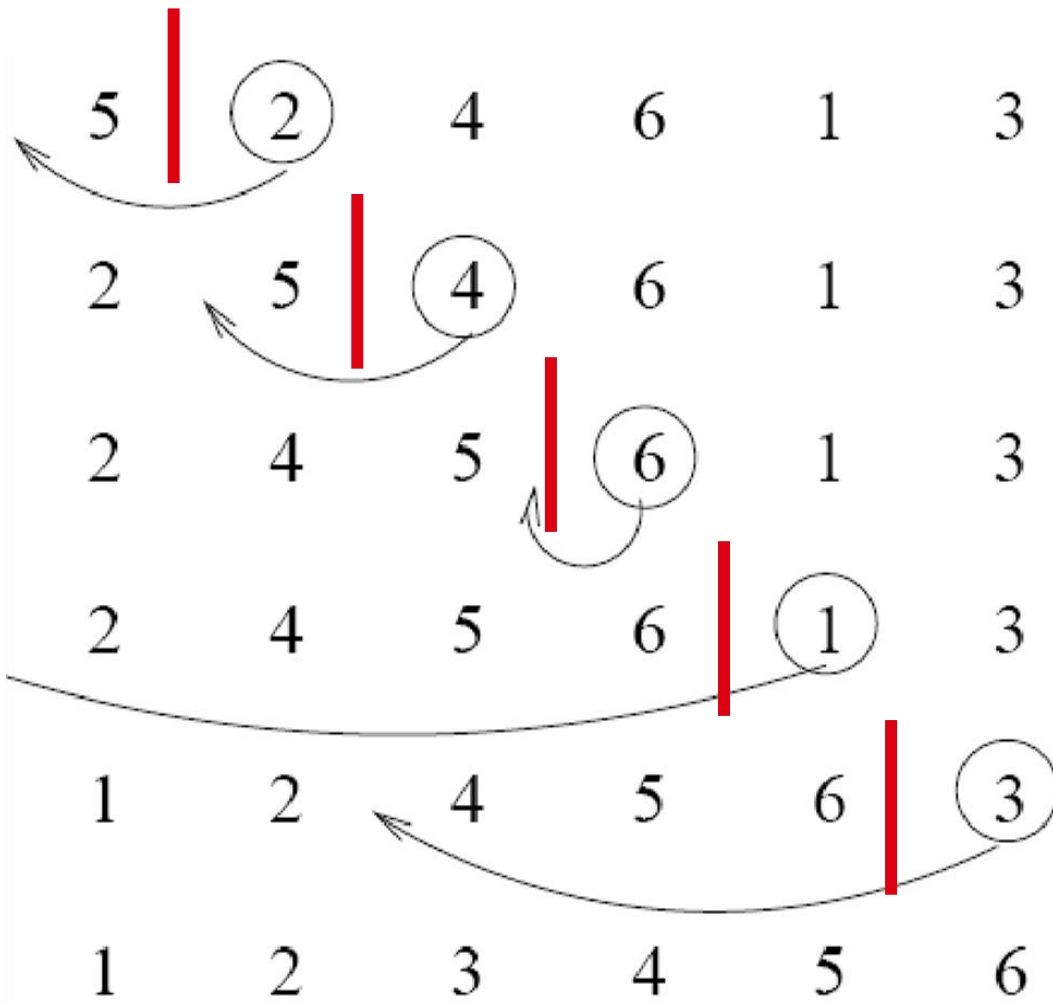




# Insertion Sort



# Insertion Sort



# INSERTION-SORT

---

```
def insertionSort(list,n):  
    for (i = 1 to n-1):  
        key=list[i]  
        j=i-1  
        while (j>=0 and list[j]>key):  
            list[j+1] = list[j]  
            j -= 1  
        list[j+1] = key
```