TECHNO INDIA UNIVERSITY
WESTBENGAL

Data Structures & Algorithms
(PCC-CS 301)

Dr. Debashis Das
Associate Professor
Department of CSE
Techno India University, Kolkata

Topics Covered

1. Data Searching
    a. Sequential search
    b. Binary search

Data Searching

- Searching of Data
  - Objective
    - Accessing of a particular data stored in a data structure
    - Searching process should be efficient in terms of speed
  - Type of searching
    - Sequential (or linear) Search
    - Binary Search
    - Hashing

    } Performed on linear data structures

    - Depth First Search
    - Breadth First Search

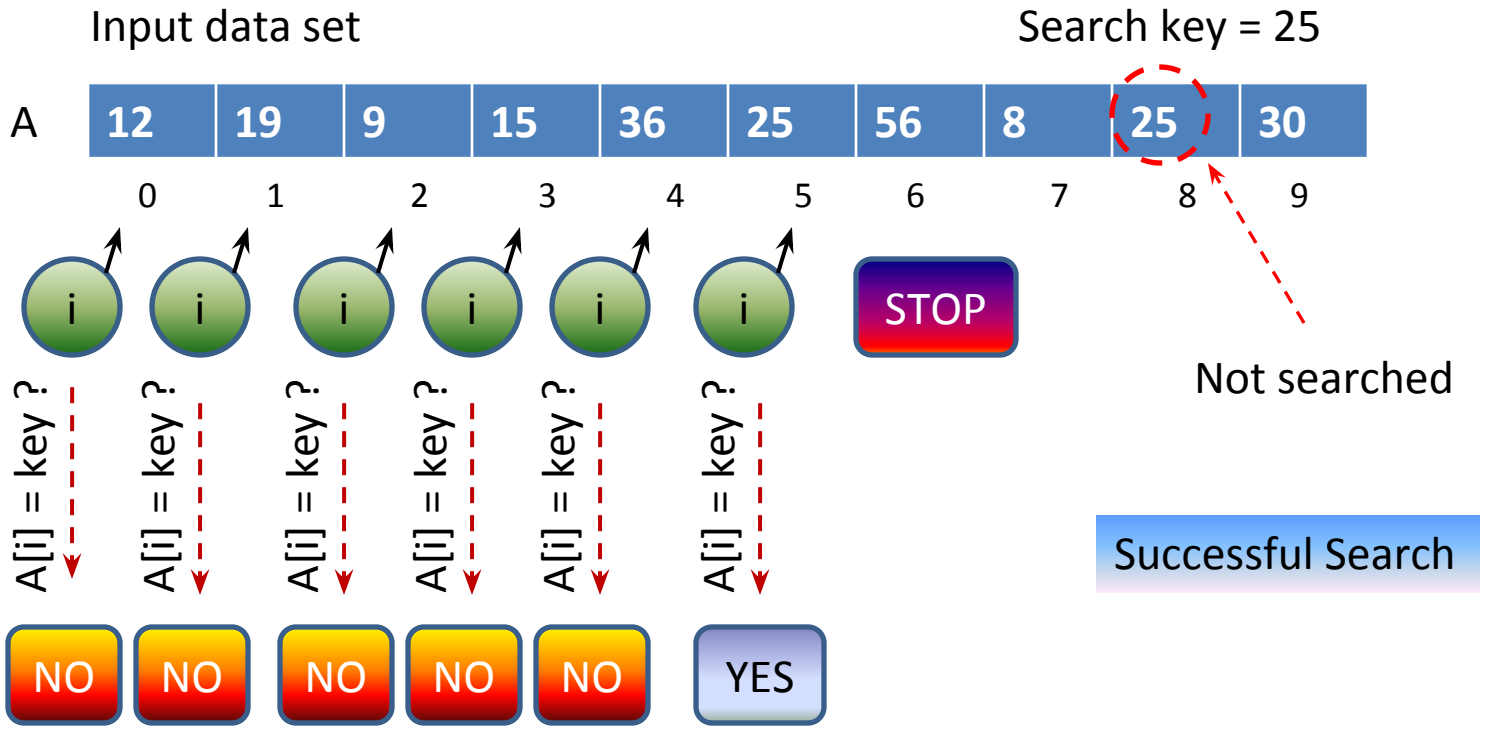
    } Performed on Graph data structures

Data Searching

- Sequential (linear) search
  - Procedural steps
    - Linear search is performed on linear data structures
    - Search process starts from the first element and ends at the last element until the given data is found
    - If the data is found in the list
      - Searching stops and return with the searched position
      - This is termed as successful search
    - If the data is not found in the list
      - Searching stops when reaches to the end of the list
      - Returns with a message as un-seccessful search

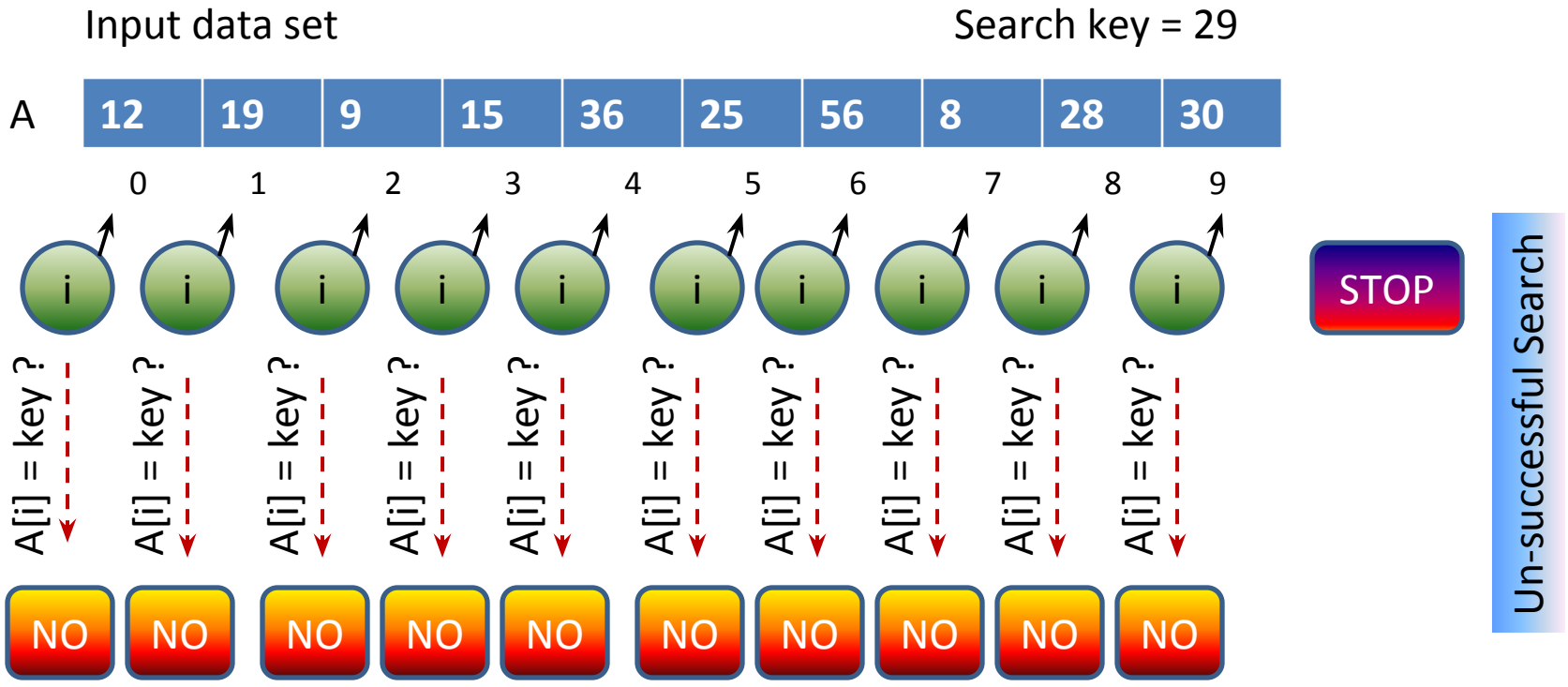
Data Searching

- Sequential (linear) search
  - Mechanism (with example)



Data Searching

- Sequential (linear) search
  - Mechanism (with example)



Data Searching

- Sequential Search
  - Algorithm

```
Linear_Search(A, key)  // A is the list, key is the data to be searched
{
  for i= 1 to n
      if  A[i] = key
         Print "successful search" and return

  Print "Unsuccessful search"
  return
}
```

Data Searching

- Sequential Search
  - Complexity analysis

We need to traverse the entire list, maximum in case of un-successful search

If there are **'n'** number of elements in the list, worst case time complexity of the process will be O(n)

In average case, if data found by traversing half of the list, execution time will be (n/2) unit i.e. Θ(n)

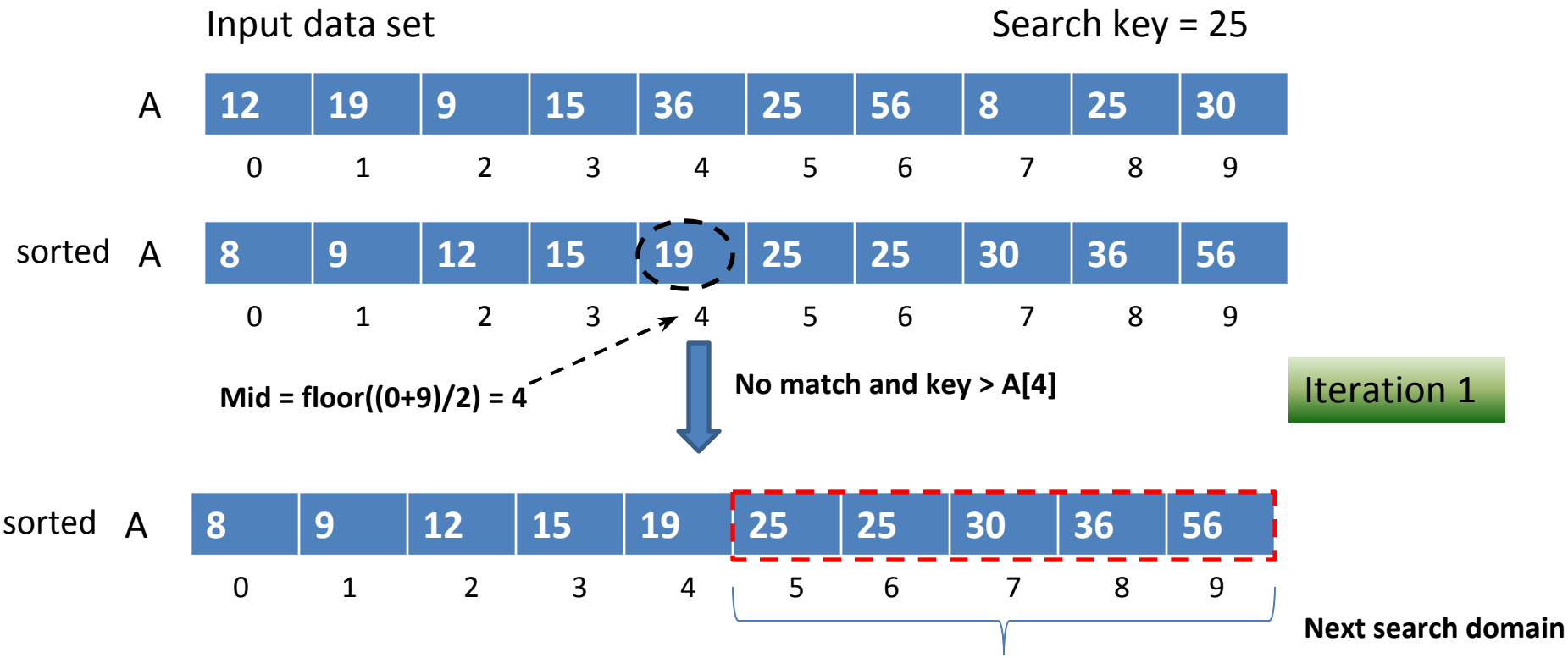
In best case, if data is searched in the first check, time complexity will be Ω(1)

Data Searching

- Binary search
  - Procedural steps
    - Binary search is performed on sorted linear data structures
    - Search process starts from the middle element of the list
    - If the key is matched with the middle element
      - Search is successful and hence stop searching
    - If the key is not matched
      - Check the key is larger or smaller than the middle element
      - If key is larger, consider the right sub part of the list as next searching domain
      - If key is smaller, consider the left sub part of the list as next searching domain

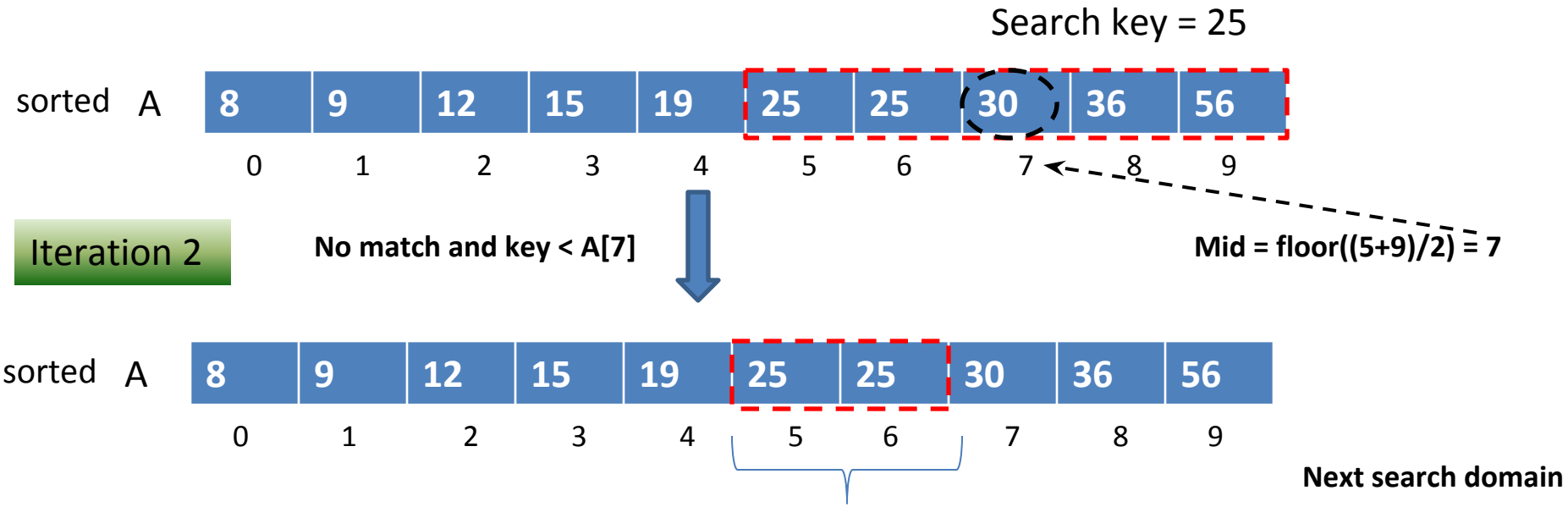
Data Searching

- Binary search
  - Mechanism (with example)



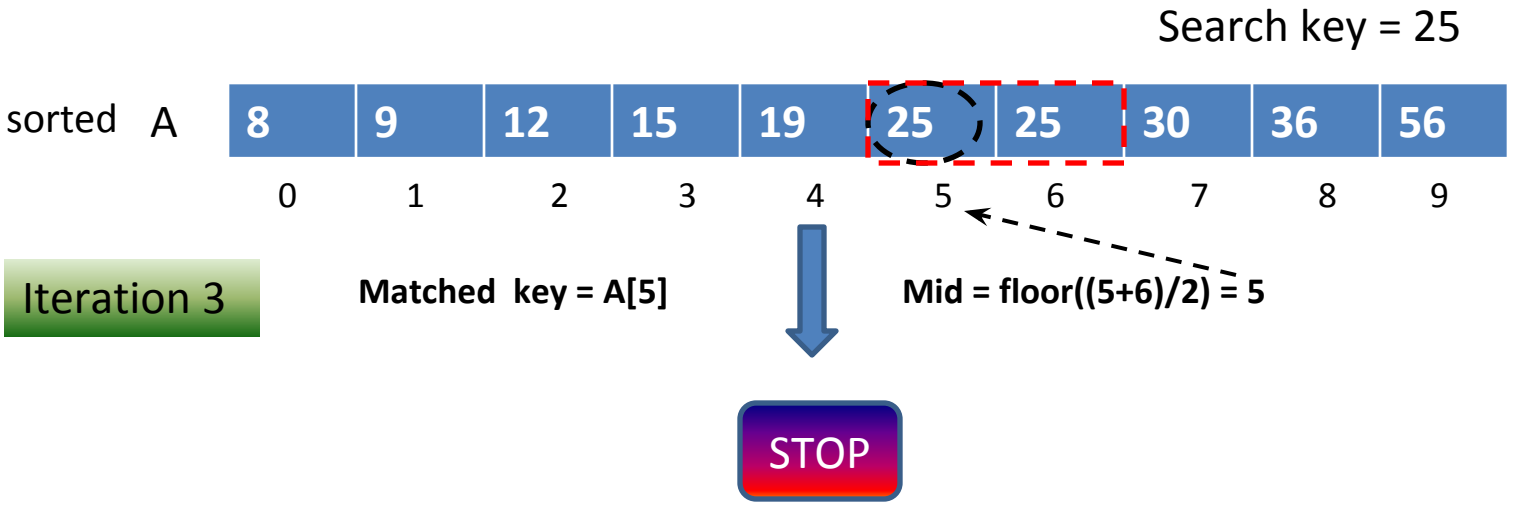
Data Searching

- Binary search
  - Mechanism (with example)



Data Searching

- Binary search
  - Mechanism (with example)



Successful Search
Data found at index 5

Data Searching

- Binary Search
  - Algorithm

```
Binary_Search(A, key)  // A is the list, key is the data to be searched
{
 set first = 0
 set last =  A.length -1  // will set the last index
 while  first <= last
       set mid = floor ((first+last)/2)
       if A[mid] = key
             print “search successful ” and return
       else if A[mid] > key
             set last = mid-1
       else
             set first = mid+1

 print “Unsuccessful search”
 return
}
```

Data Searching

- Binary Search
  - Complexity analysis

In each iteration, the searching space will be half of the previous

If in every iteration, input size is getting half, the complexity will be in logarithmic form

Hence, the worst case time complexity will be **$O(\log_2 n)$**

In best case, if data is searched in the first check, time complexity will be **$\Omega(1)$**

Queries?