



## *List ADT*

- A **list** or **sequence** is an abstract data type that represents a countable number of ordered values, where the same value may occur more than once.

A sequence of zero or more elements  $A_1, A_2, A_3, \dots, A_N$

$N$ : length of the list  $A_1$ : first element  $A_N$ : last element  $A_i$ : position  $i$

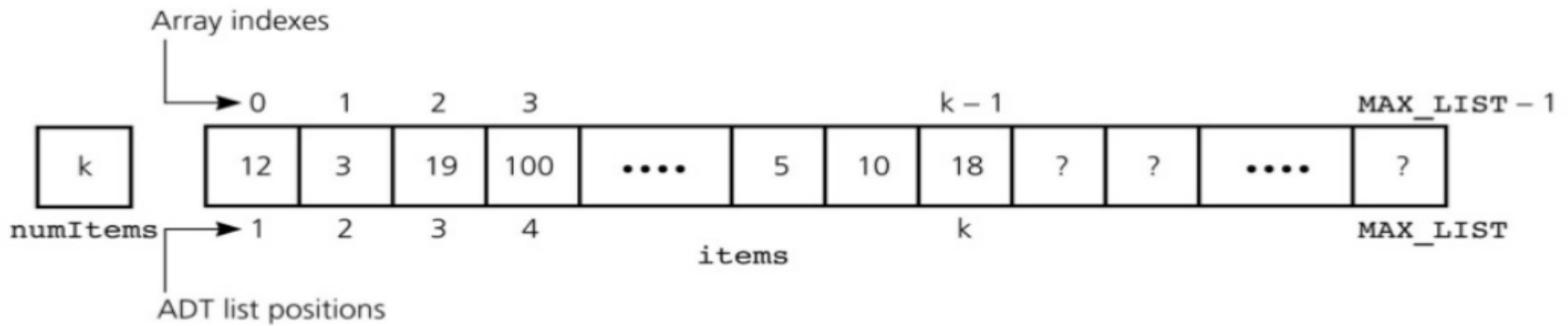
If  $N=0$ , then empty list Linearly ordered

$A_i$  precedes  $A_{i+1}$   $A_i$  follows  $A_{i-1}$

Lists are a basic example of containers, as they contain other values. If the same value occurs multiple times, each occurrence is considered a distinct item.



A list's  $k^{\text{th}}$  item will be stored in `items [k-1]`



An array-based implementation of the ADT list



# Operations





# THE LIST ADT

---

- **Example:**

the elements of a list are

34, 12, 52, 16, 12

- Find (52)  $\rightarrow$  3
- Insert (20, 3)  $\rightarrow$  34, 12, 52, 20, 16, 12
- Delete (52)  $\rightarrow$  34, 12, 20, 16, 12
- FindKth (3)  $\rightarrow$  20



# Array Implementation of List ADT

- Need to define a size for array
  - High overestimate (waste of space)
- Operations Running Times
  - PrintList }  $O(N)$
  - Find }  $O(N)$
  - Insert }  $O(N)$  (on average half needs to be moved)
  - Delete }  $O(N)$
  - FindKth }  $O(1)$
  - Next }  $O(1)$
  - Previous }  $O(1)$



# Array Implementation of List ADT

---

- Disadvantages :
  - insertion and deletion is very slow
    - need to move elements of the list
  - redundant memory space
    - it is difficult to estimate the size of array