COMP 250

Lecture 4

Array lists

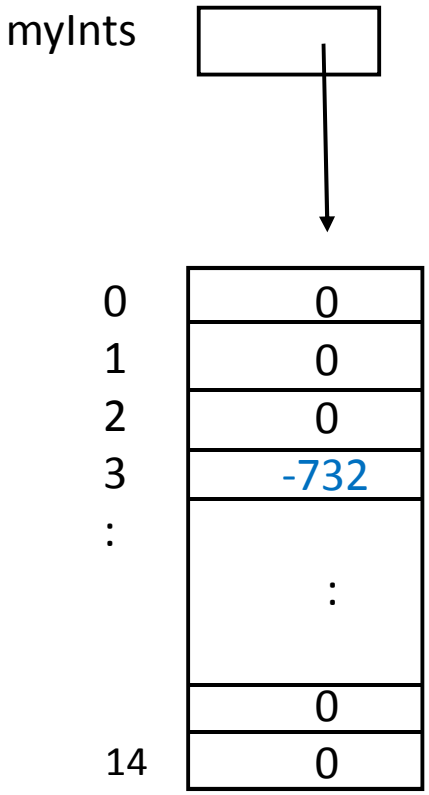
Sept. 15, 2017

Arrays in Java

```
int[ ]        myInts      =   new int[15];

              myInts[3]   =   -732;
```

Array whose elements have a *primitive* type




```
int[ ]   myInts      =  new int[15];
         myInts[3]   =  -732;
```

Arrays in Java

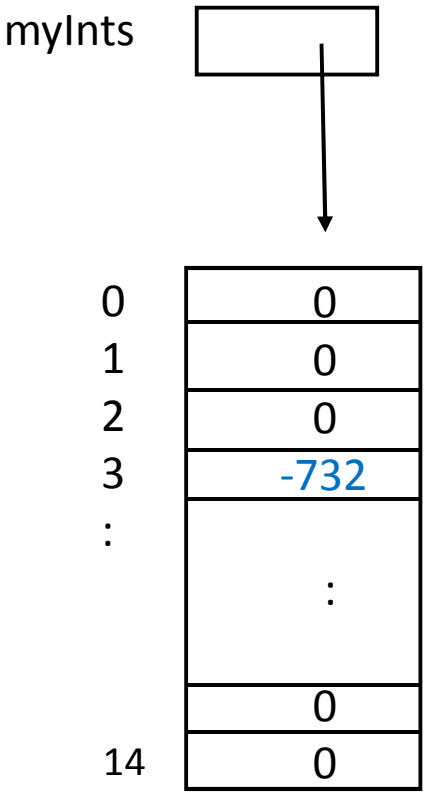
```
Shape[ ]    shapes  =  new    Shape[428];

            shapes[293] =  new Shape(    );
```

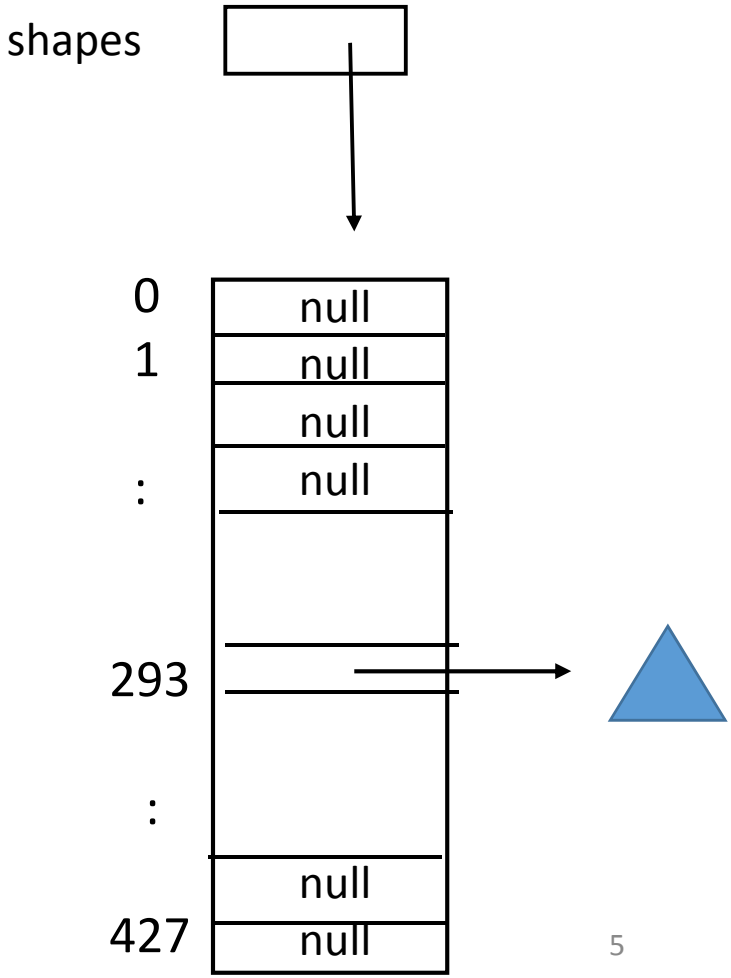
The symbol here corresponds to some arguments that specify a shape.

Array whose elements have a *reference* type

```
int[ ]    myInts      =  new int[15];
          myInts[3]   =  -732;
```

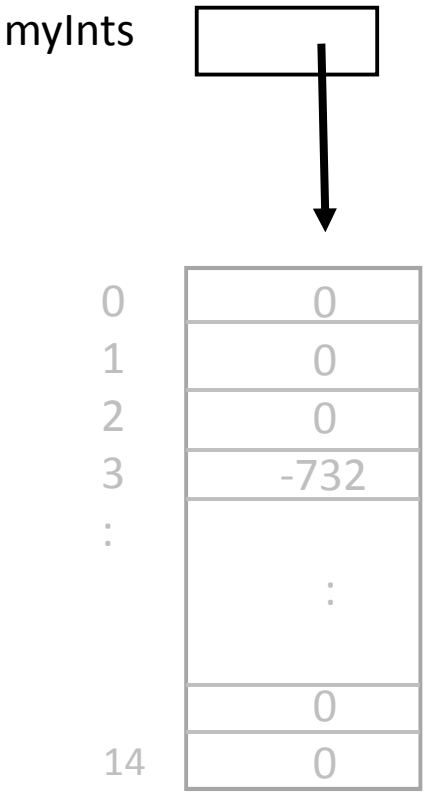


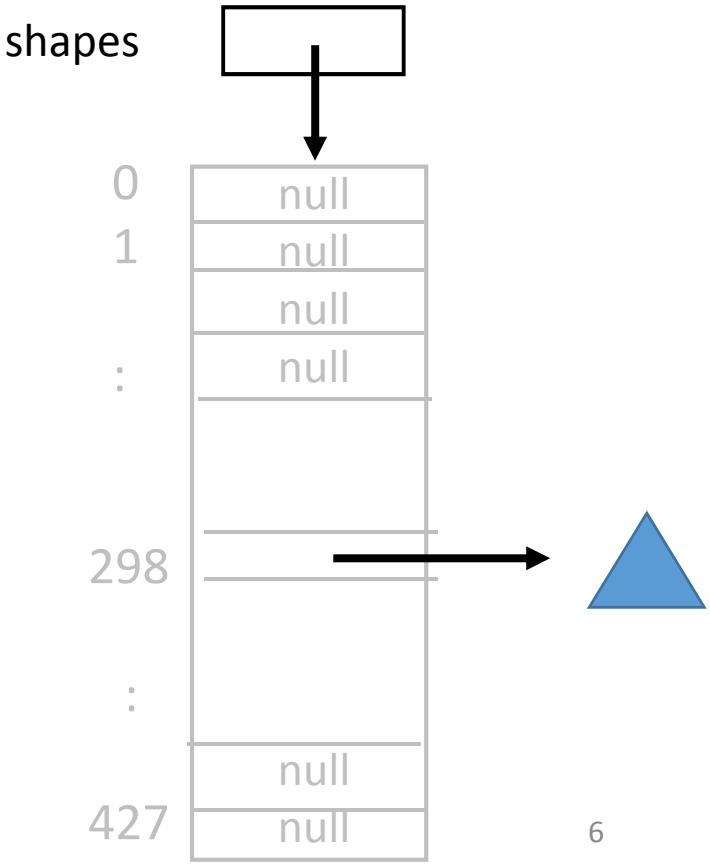

```
Shape[ ]    shapes   =  new Shape[428];
            shapes[293] =  new Shape(  );
```



The value of a reference variable is an "*address*" which specifies where an object is in the computer memory. We often represent a reference with an arrow

In the C programming language, you have access to that value and can manipulate it. In Java, you have access to it but you can't use it.





Arrays have constant time access

A computer accesses an element in an array in constant time

i.e. constant, independent of the length N of the array.

```
....     =  a[k] ;          //  read

a[k]   =    .... ;          //  write
```

You will learn more about how this works in COMP 206 and 273.

Arrays versus ‘Array Lists’

Arrays *can be used* to make lists, sometimes called ‘array lists’.

Java has an ArrayList class.

List

An ordered set of elements

$$a_0 , a_1 , a_2 , a_3 , \ldots , \ a_{N-1}$$

N is the number of elements in the list, often called the “size” of the list.

What things do we do with a list?

```
get(i)        //  Returns the i-th element (but doesn't remove it)
set(i,e)      //  Replaces the i-th element with e
add(i,e)      //  Inserts element e into the i-th position
remove(i)     //  Removes the i-th element from list
remove(e)     //  Removes first occurrence of element e
              //   from the list (if it is there)
clear()       //  Empties the list.
isEmpty()     //  Returns true if empty, false if not empty.
size()        //  Returns number of elements in the list
```

Lists

- array list (today)
- singly linked list
- doubly linked list

:

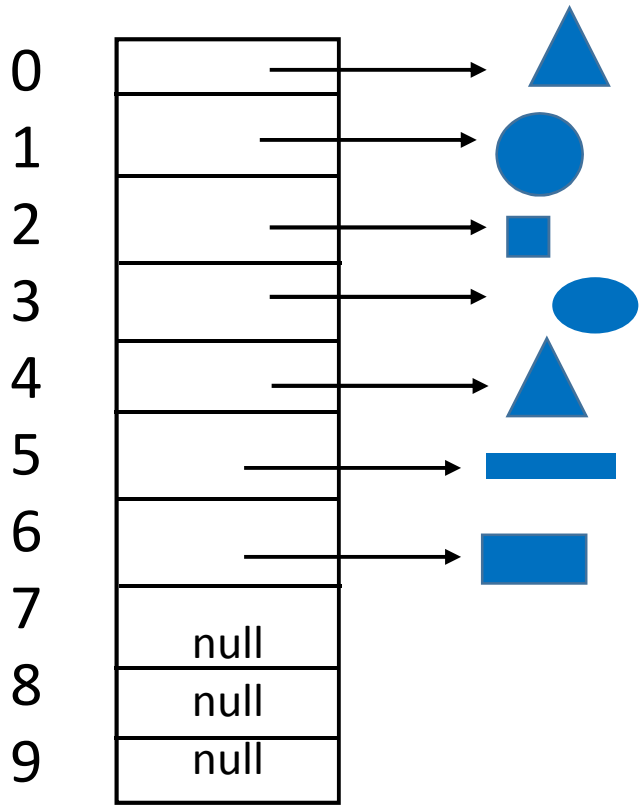
next week

array list of int

0	4
1	-3
2	19
3	-7
4	221
5	0
6	16
7	0
8	0
9	0
10	0

size = 7
length = 11

array list of Shape

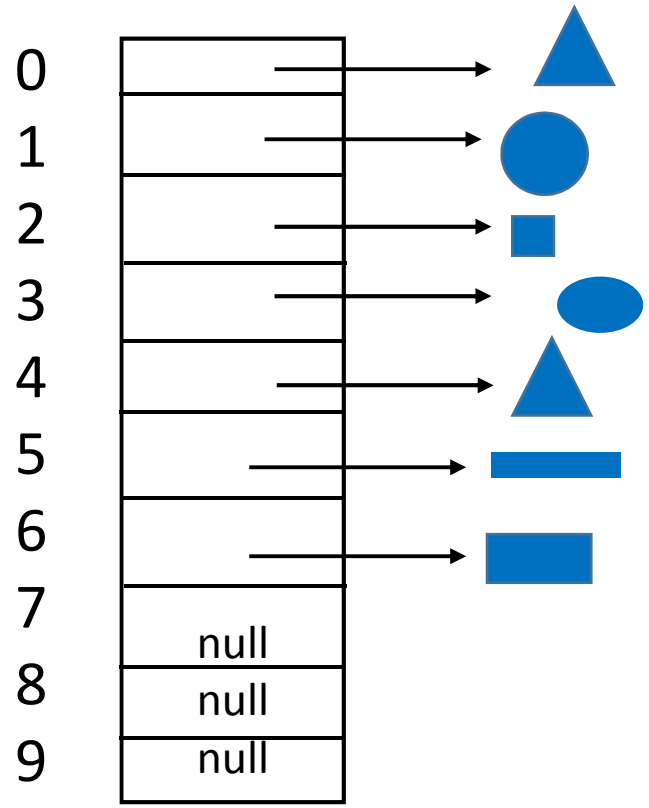


size = 7
length = 10

Let’s assume that the array is a[ ].
How to implement various operations ?

```
get(i) {

    if (i >= 0) &  (i < size)
        return a[i]
}
```

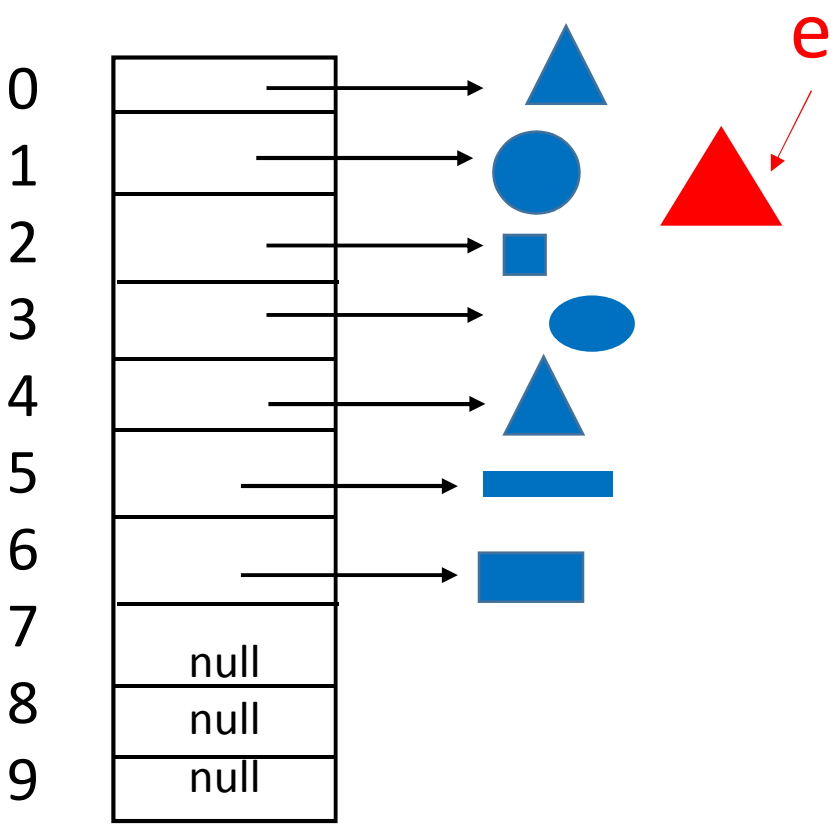


size = 7
length = 10

```
set(i,e){    // replaces the object at index i

    if (i >= 0) &  (i < size)
        a[i] = e
}
```

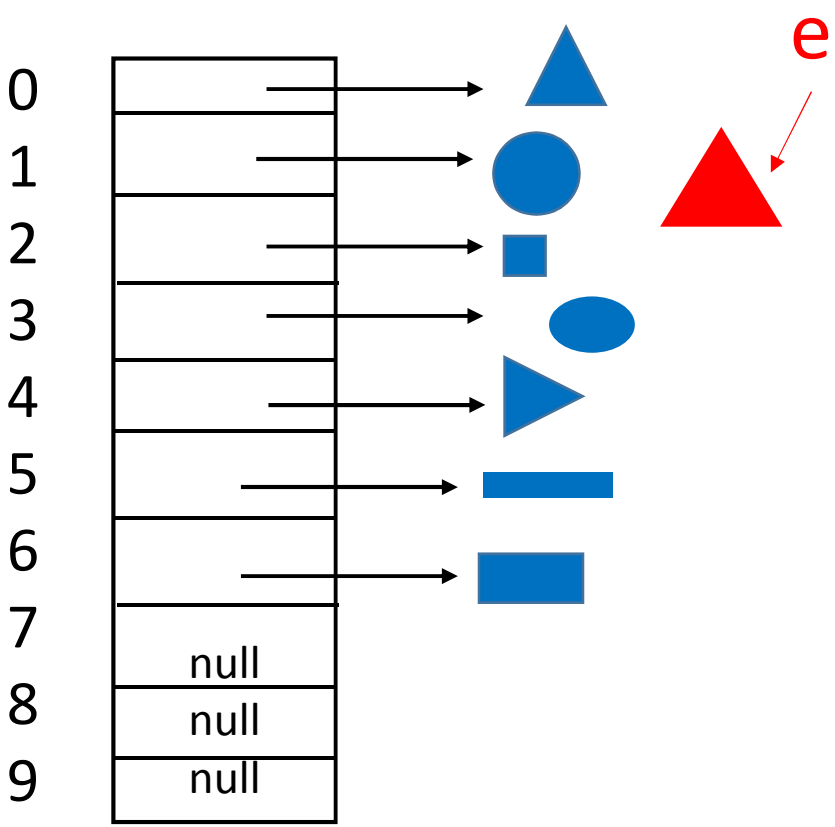
e.g. set(4, e)

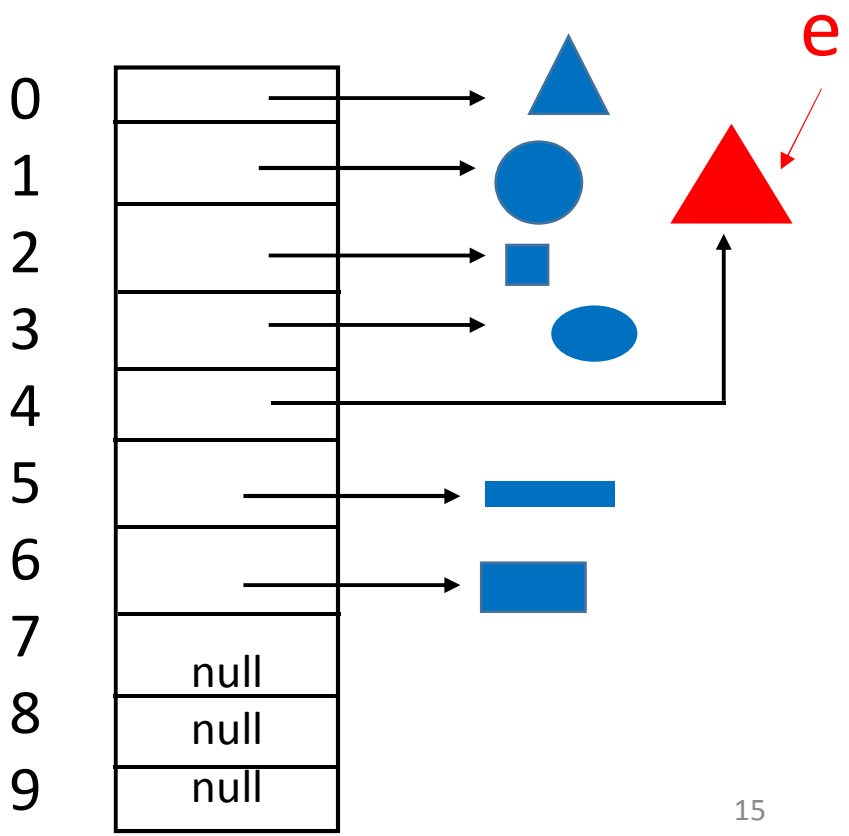


```
set(i,e){    // replaces the object at index i

    if (i >= 0) &  (i < size)
        a[i] = e
}
```

e.g. set(4, e)

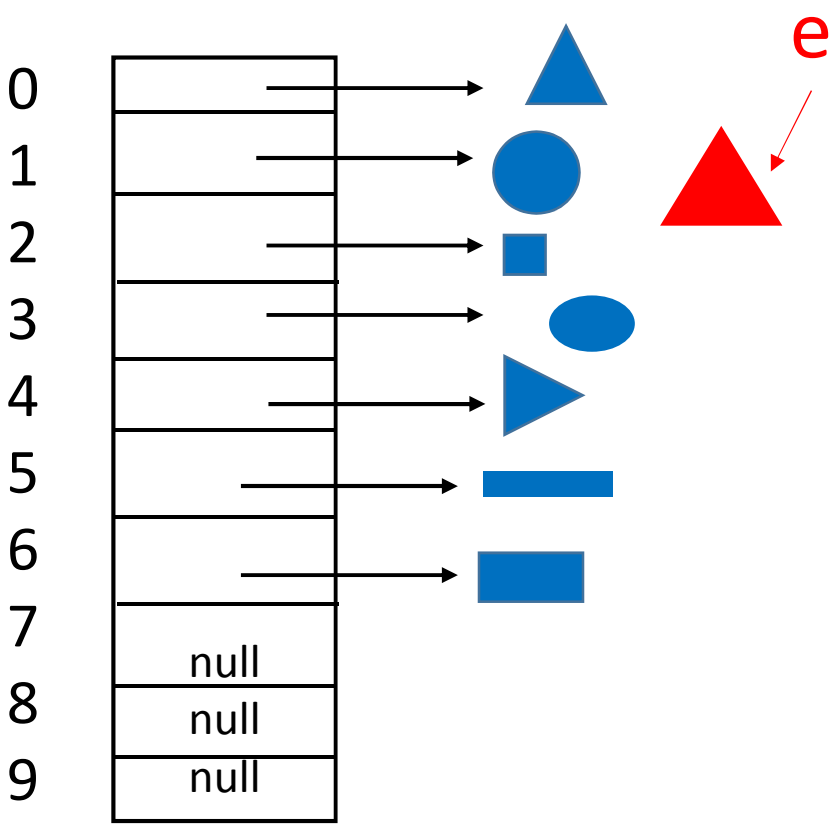




add(i, e)

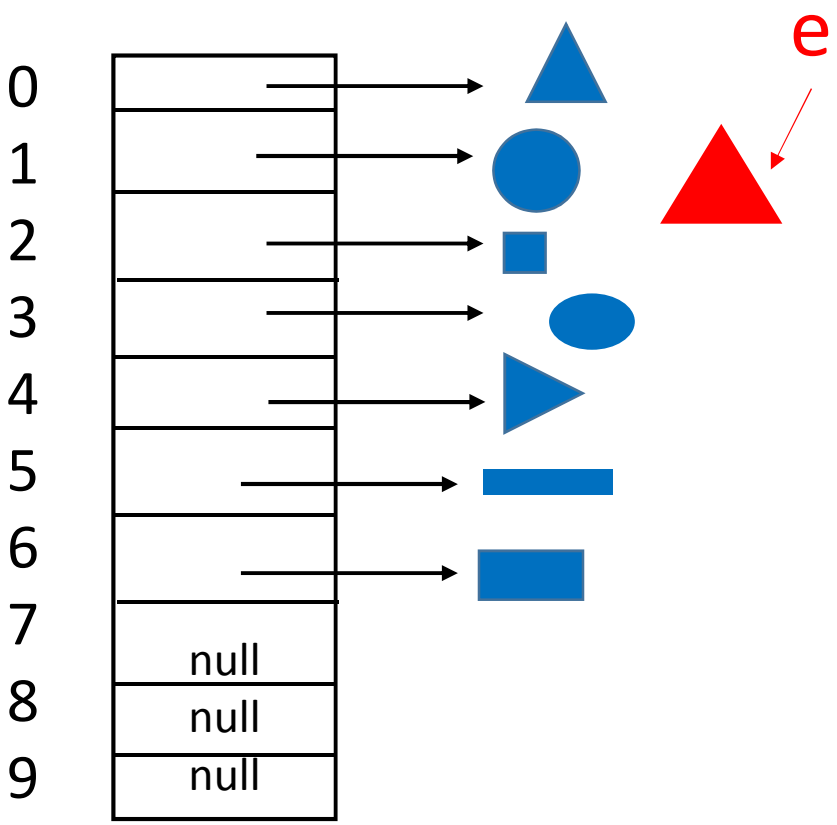
Make room by shifting, and then change reference.

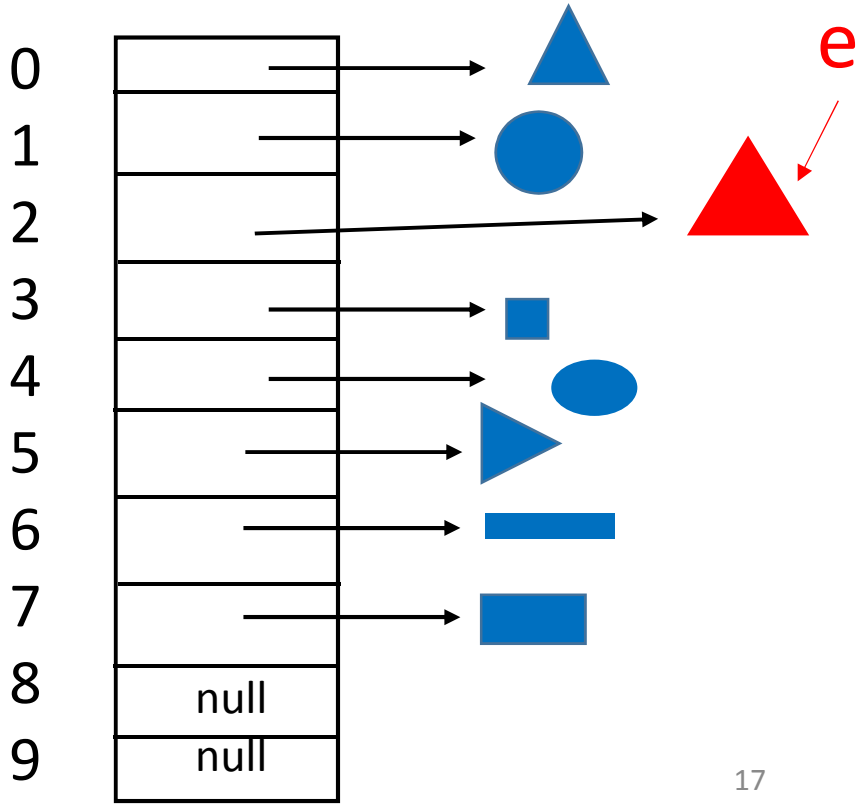
e.g. add(2, e)



add(i, e)

Make room by shifting, and then change reference.
e.g. add(2, e)



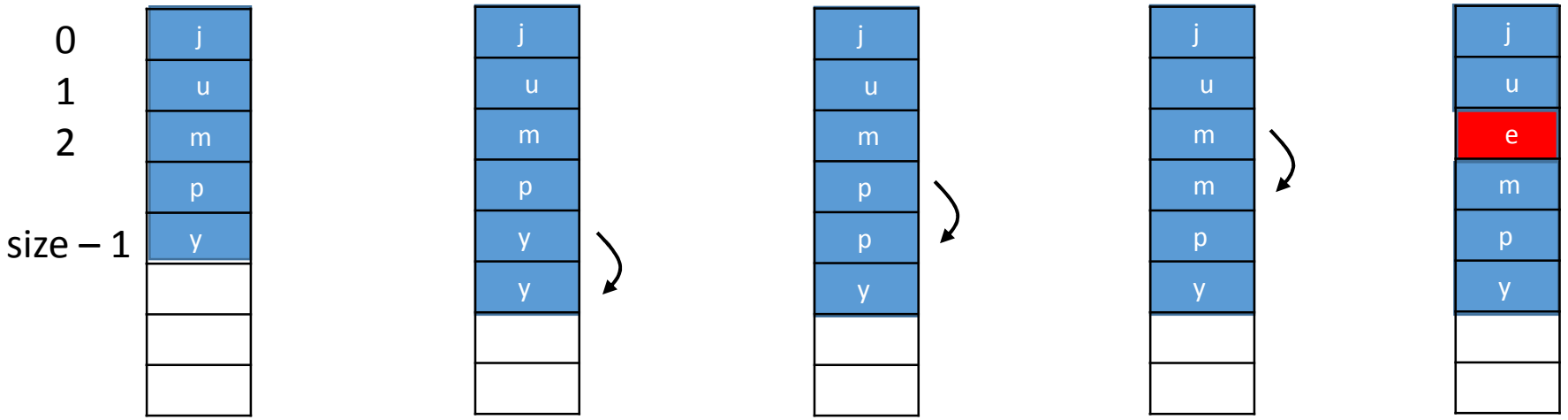


```
add( i, e) {                          //  in the figure below,  add( 2,  e)

    if   (i >=0) & (i <= size){

        for (j = size; j > i;  j--)
            a[j] = a[j-1]             //   shift (copy)

        a[i] = e                      //  replace value
        size = size + 1               //  increase number of elements
    }
}
```

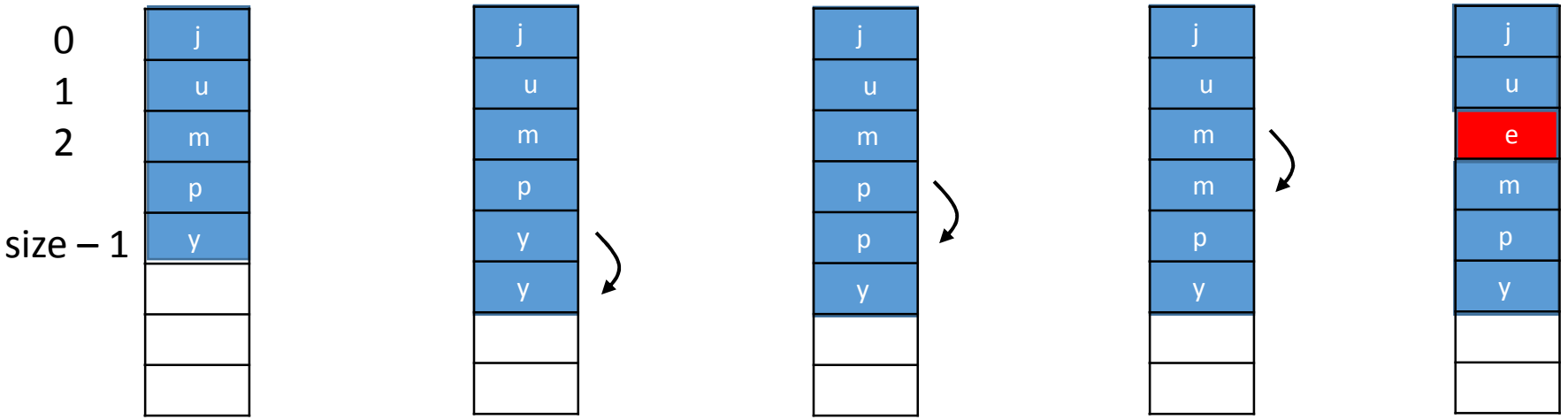


```
add( i, e) {                          //  in the figure below,  i = 2

    if   (i >=0) & (i <= size){

        for (j = size; j > i;  j--)
              a[j] = a[j-1]           //   shift (copy)

        a[i] = e                      //  replace value
        size = size + 1               //  increase number of elements
    }
}
```



How to add an element to an array list when array is full ?

```
add( i, e) {

    //  Create an empty bigger array.

    //  Copy all elements to bigger array.

    //  Add new element to the bigger array.
}
```

How to add an element to an array list when array is full ?

```
add( i, e) {

        if (a.size == a.length){          //  is array full?
          make new bigger array b         // e.g.   b.length = 2*a.length
          for ( int i=0;  i < size;  i++)
                b[i] = a[i]               // copy elements to b

          a = b
        }

      //  insert the add( i , e ) code from earlier.
}
```

SLIDE ADDED

What if you want to add an element to the list because you don't care where it goes?

Or what if you want to add an element to the end of the list?

The add(i, e) code does not allow this.
Instead we need another method add(e).

See Exercises.

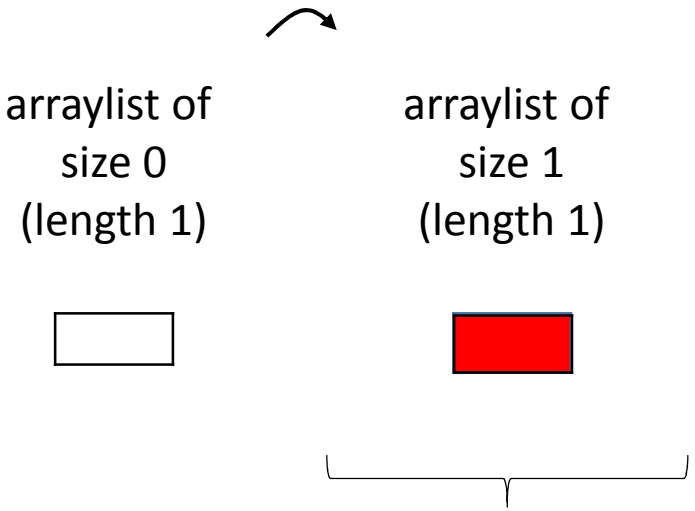
Overloading

```
add( e )      //  inserts element e at end of list
add( i ,e)    //  Inserts element e into the i-th position


remove(i)     //  Removes the i-th element from list
remove(e)     //  Removes first occurrence of element e
              //   from the list (if it is there)
```

Adding N elements to an array list

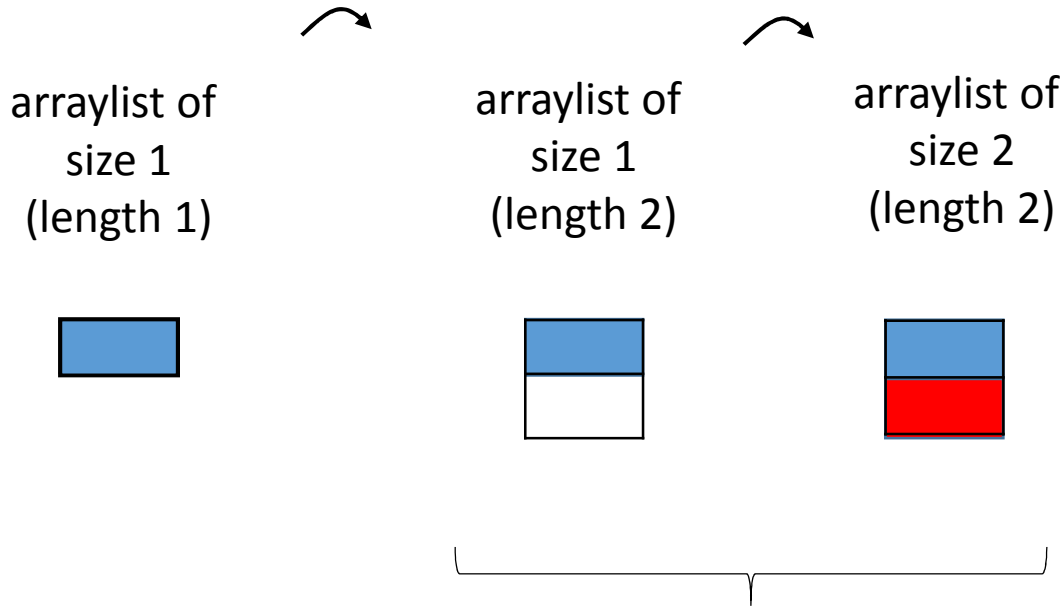
Suppose we initialize an array list with an empty array of length 1.
We then add an element.



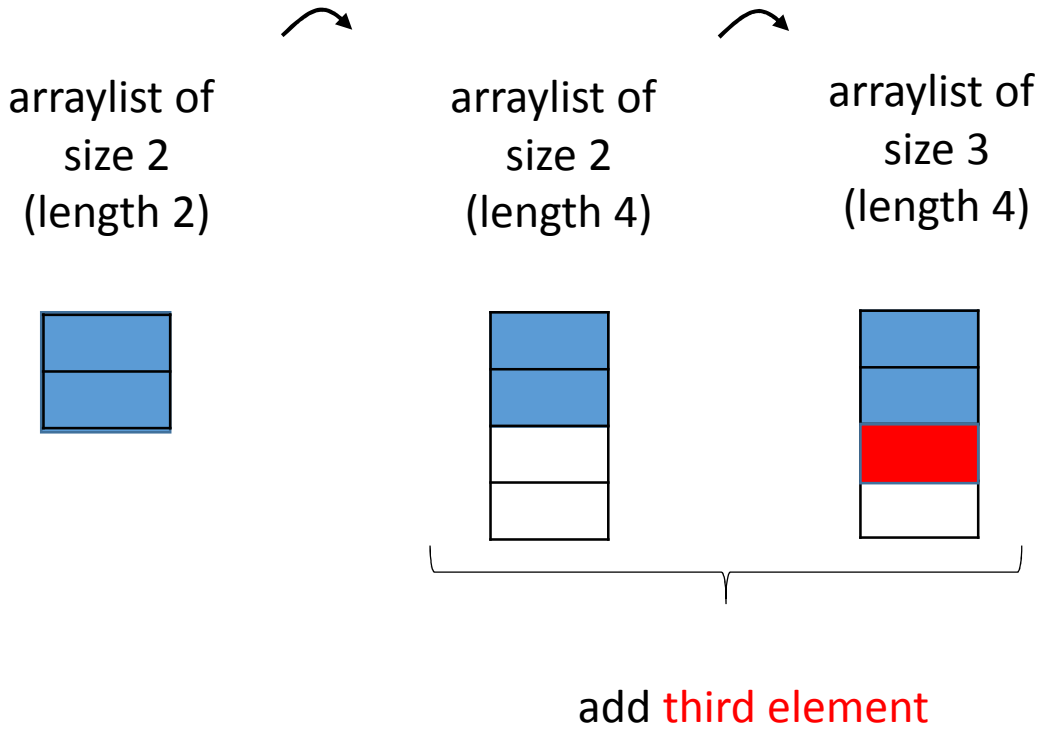
What do we do to add a second element?

Adding N elements to an array list

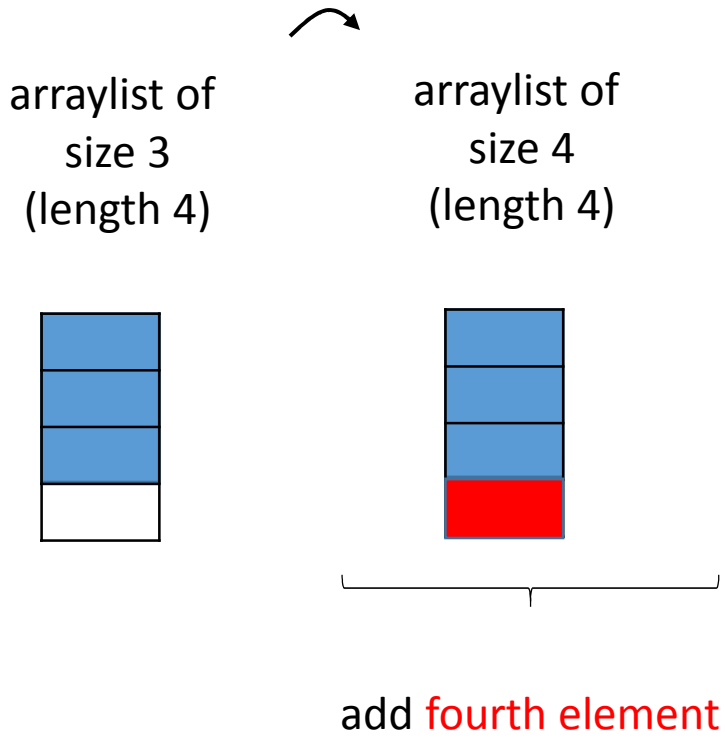
Suppose each time we add to a full array list, we double the length of the array.



Adding N elements to an array list



Adding N elements to an array list

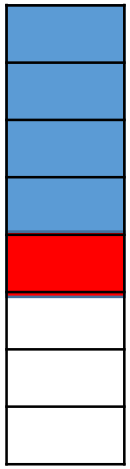


Adding N elements to an array list

arraylist of
size 4
(length 4)

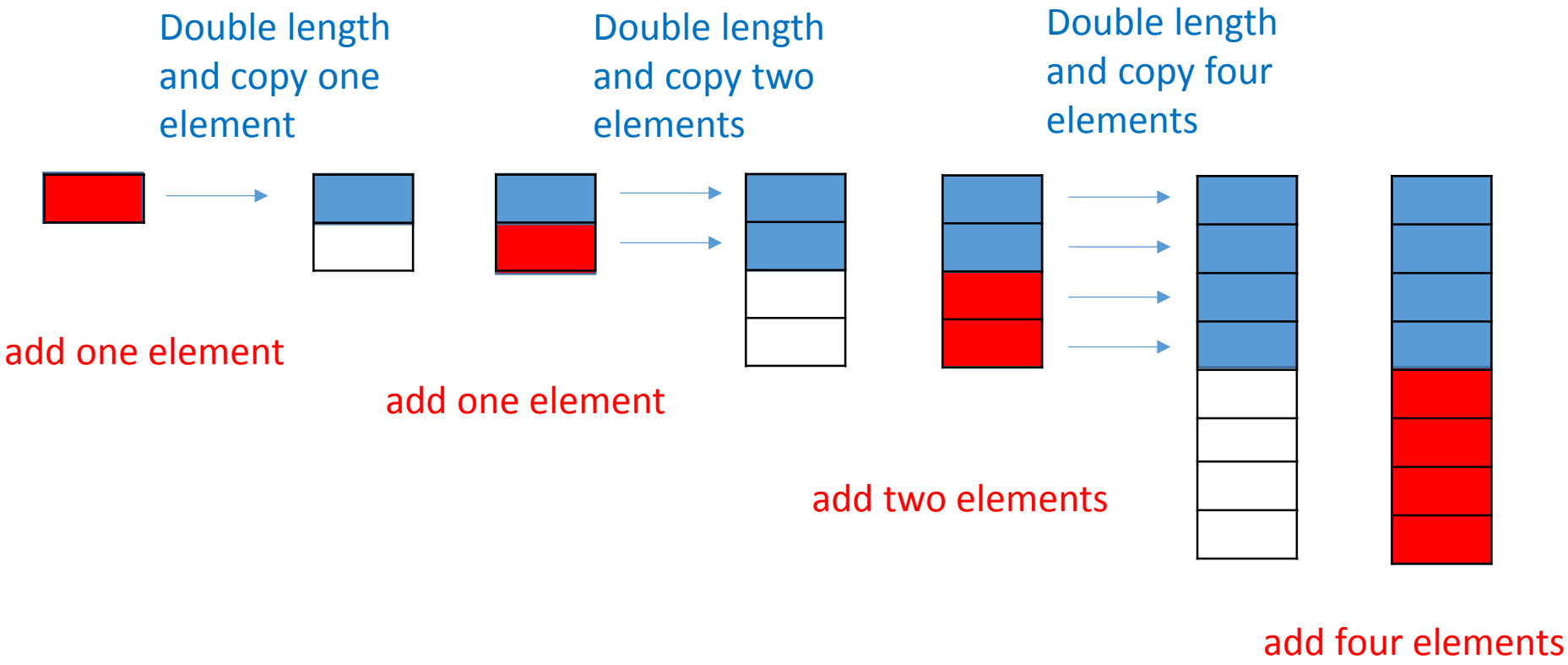
arraylist of
size 4
(length 8)

arraylist of
size 5
(length 8)



add fifth element

Adding N elements to an array list



Q: How many times k do we need to double the length of the array so that it is of length N ?

A:

Q: How many copy operations are required to add N elements to an empty array list ?

A:

Q: How many times k do we need to double the length of the array so that it is of length N ?

A: $2^k = N, \quad so \quad k = log_2 N$

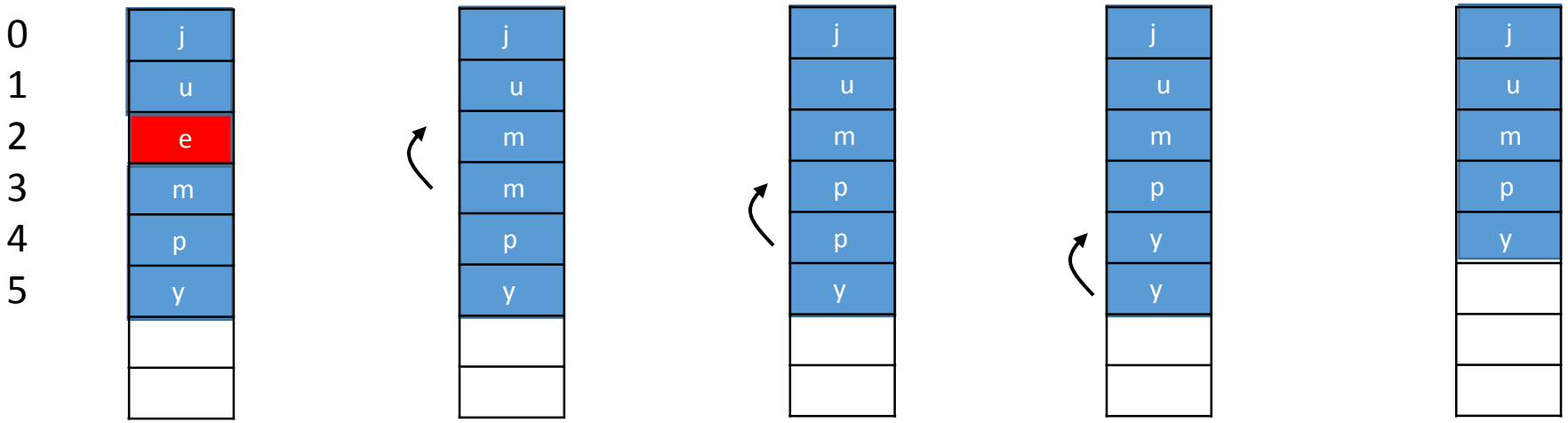
Q: How many copy operations are required to add N elements to an empty array list ?

A: $1 + 2 + 4 + 8 + \ldots 2^{k-1} = 2^k - 1 = N - 1$

List Operations

```
get(i)
set(i,e)
add(i,e)
remove(i)       //  Removes the i-th element from list
remove(e)       //  Removes element e from the list (if it is there)
clear()         //  Empties the list.
isEmpty()       //  Returns true if empty, false if not empty.
size()          //  Returns number of elements in the list
  :
```

remove(i) // in the figure below, i = 2



size = 6

size = 5

remove(i)

```
if ( (i >= 0)  and  (i < size) ){

     tmp = a[i]                          //    put aside and later  return it

    for ( k = i;  k < size-1;  k++){
      a[ k ] = a[ k + 1 ]                        //    shift (copy)
    }

    size = size – 1
    a[ size ] = null                     //    clean
    return tmp
}
```

Quiz 0 : Test your Java skill

- Worth 0% of your grade
- Starting today at noon until Monday night
- Practice mycourses/quiz mechanism and timing
- Allow us to test if the system works as we think
- Allow you/us to calibrate