Insertion & deletion of array

[1, 5, 3, 3] → [1, 3, 3,

deletes

A) In the end

N = 6 [1, 5, 3, 2, 3, 10, ...]

Delete

N = 5 [1, 5, 3, 2, 3]

Insert 21

N = 7

[1, 5, 3, 2, 3, 10, 21]

B) In the beginning

N = 6 [1, 5, 3, 2, 3, 10, ....]

⇓ delete

[5, 3, 2, 3, 10]

~~arr[0] = 0~~;

arr[1] = arr[0]

arr[0] = 0;

if N == 0
    return N;

N = 6

[1, 5, 3, 2, 3, 10, ...]

⇓ Insert 4

[4, 1, 5, 3, 2, 3, 10]

c) At any given index

delete ⇒ i ∈ [0, N-1]

shift left (deletion)

[3], [1], [5], [8] ⇒ subarrays of size 1 (5)

Insertion & deletion   Day-11

## Day-11

→ Insertion & Deletion in an Array

```
import java.util.Scanner;
class ClassA{
  public static void display (int[] arr, int N){
   for(int i=0; i<N; i++){
        S.o.p( arr[i] + " ");
        S.o.pln(" ");
   }
  }
 public static void main(String[] args){
  Scanner sc = new Scanner(System.in);
  int arr[] = new int[100];
   int N = sc.nextInt();
  for (int i=0; i<N; i++)
       arr[i] = sc.nextInt();
   display(arr,N);
   // Delete from end
     if(N>=1){
       ~~N~~
      arr[N-1]=0;
       N--;
   //Insert 15 in the end
     if(N<100){
       arr[N]=15;
        N++;
     }

                              N = delete_from_end(arr,N);
                              display(arr, N);

    public static int insert_at_end (int[] arr, int N, int val){
      if(N<100){
         arr[N]=val;
          return N+1;
      }
     else return N;
    }
public static int delete_at_end (int[] arr, int N){
              if(N>=1){
          arr[N-1]=0;
              return N-1;
          }
        else return N;
```

```
public static int delete_from_begining (int[] arr, int N){
  if ( N==0)
     return N;
  for(int i=0; i+1<N; i++) {
        arr[i] = arr[i+1];
     }
     arr[N-1]=0;
     return N-1;
  }

public static int insert_at_begining (int[] arr, int N, int val){
  if(N==arr.length)
     return N;
  for (int i=1; i<N; i++) {
    arr [i] = arr[i-1];
  }
  arr[0]= val;
   return N+1;
  }

public static int delete_at_index (int[] arr, int N, int id) {
 if (N==0)
   return N;
 if(id<0 ||id>=N)
     return N;
 for(int i=id; i+1< N; i++) {
   arr[i]= arr[i+1];
   }
  arr[N-1]=0;
  return N-1;
 }
public static int insert_at_index (int[] arr, int N, int id,
                                                int val ) {
  if(N==arr.length)
       return N;
   if(i<0 || i>N)
      return N;
 for(int ~~i=N; i>=~~ i=id+1; i<=N; i++){
     arr[i]=arr[i-1];
   }
    arr[id]=val;
       return N+1; }
```