For Data Access Array is recommend ::
Time Complexity :
If we know the index number then its = O(1)
Otherwise Iteration is needed so O(n)

Space Complexity : O(n)

It's On Array
Array is fixed in size.

int[] a = new int[5];

Data Deletion =>
Time Complexity :
when you delete an element all the elements should be moved an index before.
It's a costlier operation.
Time Complexity : O(n)

For Data Deletion Array is not recommended.

4000addr = 10
4004addr = 20
4008addr = 30
4012addr = 40

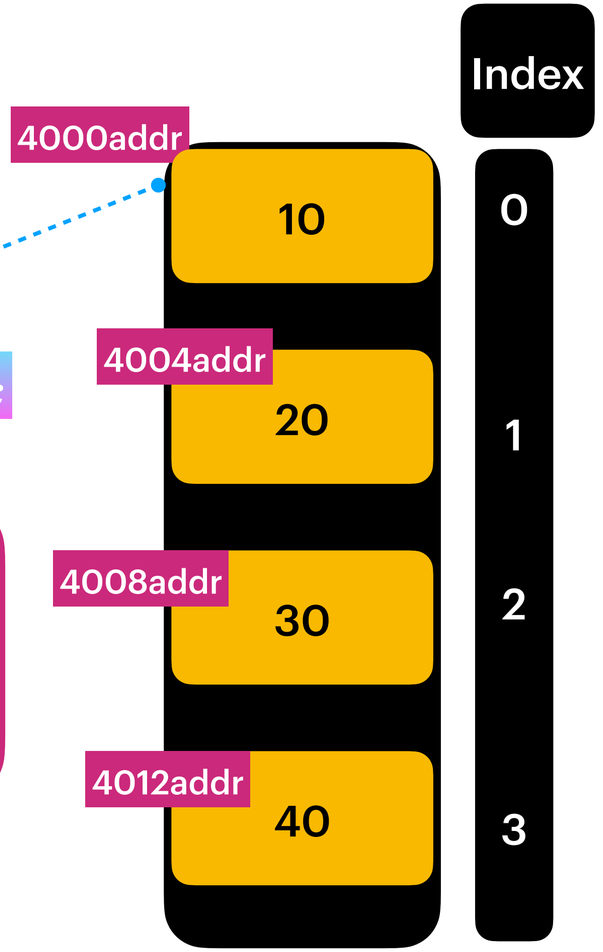
int[] a = {10,20,30,40};

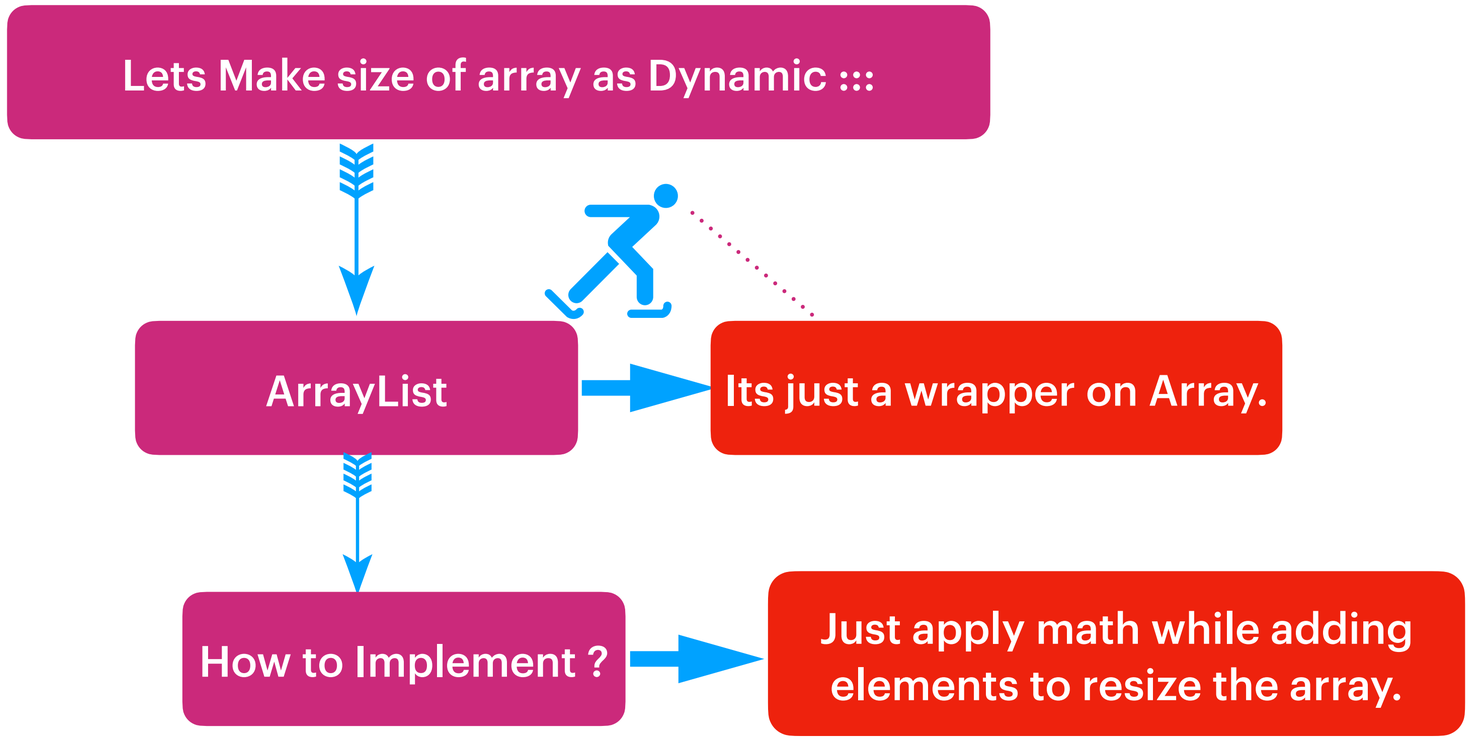
a[0] = 4000 + 0*4 = 4000 addr = 10

a[1] = 4000 + 1*4 = 4004 addr = 20

a[2] = 4000 + 2*4 = 4008 addr = 30

a[3] = 4000 + 3*4 = 4012 addr = 40




Lets Make size of array as Dynamic :::
ArrayList
Its just a wrapper on Array.
How to Implement ?
Just apply math while adding elements to resize the array.

About ArrayList :

=> Dynamic in Size .
=> Indexbased, maintains insertion order.
=> Accesibility is faster.

Drawback :
=> Allows Duplicate elements.
=> When we remove elementstakes O(n) shifts. This is constlier operation.

	Time Complexity	Space Complexity
Add Element	O(1)	O(n)
Remove Element	O(n)	O(1)
Search Element	O(n)	O(1)
Get Element based On Index	O(1)	O(1)

Go for ArrayList only when you try to access the elements.

ArrayList is not recommended when you try to remove elements frequently .