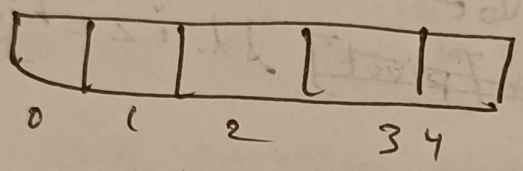
Arrays

Basics of Array → ds → similar data types.

char
int
pair<int, int>

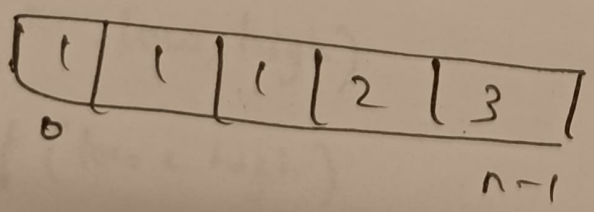
arr[] ⇒ 

int arr[6]; // stored with garbage value.
if it's inside int main()

int arr[6]; // it will be '0' if it's global.

Max size of array int arr[10^6] // inside int main()

int arr[10^7] // Globally.




```
for (i=0; i<n; i++)
{
}
    print (arr[i]);
```

Q) Largest Element in an array

arr[] = {3, 2, 1, 5, 2}

o/p = 5

Brute force

sort ⇒ (1, 2, 2, 3, 5)

print (arr[n-1]) → largest

T.C ⇒ O(NlogN) S.C ⇒ O(1)

Optimal

largest = a[0] → 5

{3, 2, 1, 5, 2}

largest = a[0]

for(i = 0 ⟶ n)

if (a[i] > largest)

largest = a[i]

cout << largest

T.C ⇒ O(N)

S.C ⇒ O(1)

Q Second largest element in arr

arr[] = {1, 2, 4, 7, 7, 5} o/p = 5

B.F

↓ Sort

1 2 4 5 7 7 largest = arr[n-1]

↑
2nd largest

```
for(i=n-2; i>=0; i--)
{ if(arr[i] != largest)
  {
    second = arr[i]
    break;
  }
}
```

T.C ⇒ NlogN + N

2nd App [1, 2, 4, 7, 7, 5]

largest = 7 2 4 7 7

→ Find the largest first (before solution) ⇒ O(N)

→ slargest = -∞ (or INT_MIN if -ves are there in arr)

7 2 4 5

slargest = -1

for(int i=0 ⟶ i++)

{ if (arr[i] > slargest && arr[i] != largest)

slargest = arr[i]

}

print(slargest) ⇒ O(N)

T.C ⇒ O(N+N) ⇒ O(2N)

Optimal App

arr[] = { 1, 2, 4, 7, 7, 5 }

largest = arr[0], slargest = ~~-1~~ (INT_MIN)

~~7 ≠ 7~~ ~~-1~~ ~~2 ≠ 1~~ ~~4 ≠ 2~~ ~~7 ≠ 4~~ ~~7 ≠ 7~~ ~~5 ≠ 7~~ (5)

if someone become largest, then prev largest will be slargest

~~for~~

Code

```
int secondLargest(vector<int> &a, int n){
    int largest = a[0];
    int slargest = -1;
    for(int i=1; i<n; i++){
```

```
// If the current Element is > than the current largest Elem
                                                        ent
if (a[i] > largest) {
    // Update the slargest element to the current
       largest element
    slargest = largest;
    // Update the largest element to the current element
    largest = a[i];
}
// If the current element is smaller than the current
// largest element and greater than the current largest
// element
else if (a[i] < largest && a[i] > slargest)
    slargest = a[i];
}
return slargest;
}
```