

Q6 Union and Intersection of two arrays.

(i) Union of arrays

i/p → {85, 25, 1, 32, 54, 63}

{85, 23}

o/p → {1, 2, 6, 25, 32, 54, 853}

↳ unique elements

As we are told to have unique elements in the o/p, then we make the use of set here. The following are the 3 steps we need to follow to find union of 2 arrays.

- 1) Copy all elements of first array to set.
- 2) Copy all elements of second array to set.
- 3) Copy the set elements to the ans array.

Dry run

- 1) $i=0$ → set
unique → {85}
- 2) $i=1$
unique → {25, 85}
- 3) $i=2$
unique → {1, 25, 85}
- 4) $i=3$
unique → {1, 25, 32, 85}
- 5) $i=4$
unique → {1, 25, 32, 54, 853}
- 6) $i=5$
unique → {1, 6, 25, 32, 54, 853}

We are done with step-1. moving on to
step-2

$$リ \mu^{\circ} = 0$$

unique $\rightarrow \{1, 6, 25, 32, 54, 853\}$

$$2) \quad i = 1$$

unique $\rightarrow \{1, 2, 6, 25, 32, 54, 85\}$

next → vector

We are done with Step-2. moving on to step-3.

1) ans → {1, 2, 6, 25, 32, 54, 853}

We have got the union of 2 arrays now.

Code

```
void findUnion (vector<int> &numsl, vector<int> &numsr) {
    // Set is used to maintain uniqueness
    set<int> unique;
    vector<int> ans;
    // Step-1
    for (int i = 0; i < numsl.size(); i++) {
        unique.insert (numsl[i]);
    }
    // Step-2
    for (int j = 0; j < numsr.size(); j++) {
        unique.insert (numsr[j]);
    }
    // Step-3
    for (auto &it : unique) {
        ans.push_back (it);
    }
    // printing ans
    for (auto i : ans) {
        cout << i << " ";
    }
}
```

(ii) Intersection of 2 arrays

I/p $\rightarrow \{1, 2, 2, 1\}$ $\{2, 2\}$ O/p $\rightarrow \{2\}$

As we are told to have unique elements only in the final ans, we will be using the set. We have to follow the following steps to find the intersection of 2 arrays.

- 1) Sort both the arrays in ascending order
- 2) Run while loop until both indexes are valid indexes.
 - $\rightarrow \text{nums1}[i] > \text{nums2}[j] \Rightarrow j++$
 - $\rightarrow \text{nums1}[i] < \text{nums2}[j] \Rightarrow i++$
 - $\rightarrow \text{nums1}[i] == \text{nums2}[j]$
 - * Insert element in set.
 - * $i++, j++$
- 3) Copy set elements to the vector named ans.

Dry run

- 1) Sort the arrays in ascending order

 $\text{nums1} \rightarrow \{1, 1, 2, 2\}$
 $\text{nums2} \rightarrow \{2, 2\}$

- 2) Run loop until $i < \text{nums1.size()} \& j < \text{nums2.size()}$.

* $\text{nums1}[i] < \text{nums2}[j] \rightarrow i++$

$\{1, 1, 2, 2\}$

$\{2, 2\}$

* $\text{nums1}[i] < \text{nums2}[j] \rightarrow i++$

$\{1, 1, 2, 2\}$

$\{2, 2\}$

* $\text{nums1}[i] = \text{nums2}[j] \rightarrow i++, j++$ but
first insert $\text{nums1}[i]$ or $\text{nums2}[j]$ into
set.

unique $\rightarrow \{2\}$

$\downarrow i$

$\{1, 1, 2, 2\}$

$\{2, 2\}$

$\uparrow j$

* $\text{nums1}[i] = \text{nums2}[j] \rightarrow i++, j++$ but
first insert the element into set.

unique $\rightarrow \{2\}$

$\downarrow i$

$\{1, 1, 2, 2\}$

$\{2, 2\}$

$\uparrow j$

Now we are done with step-2 as i & j
are not valid indexes now.

3) Copying set elements to the ans vector.
ans $\rightarrow \{2\}$

↳ Intersection of 2 arrays.

Code

```
void findIntersection (vector<int>& nums1,  
                      vector<int>& nums2){
```

```
//Step-1
sort(nums1.begin(), nums1.end());
sort(nums2.begin(), nums2.end());
vector<int> ans;
set<int> unique;

//Step-2
int i=0;
int j=0;
while (i < nums1.size() && j < nums2.size())
{
    if (nums1[i] > nums2[j]) {
        j++;
    }
    else if (nums1[i] < nums2[j]) {
        i++;
    }
    else {
        unique.insert(nums1[i]);
        i++;
        j++;
    }
}

(auto& it : unique) { //Step-3
    ans.push_back(it);
}

(auto i : ans) { cout << i << " " ; } } } } }
```