

Q19. Common elements in 3 sorted arrays.

i/p $\rightarrow \{1, 5, 10, 20, 40, 80\}$

$\{6, 7, 20, 80, 100\}$

$\{3, 4, 15, 20, 30, 70, 80, 120\}$

o/p $\rightarrow 20 \ 80$

We will simply traverse all the arrays simultaneously with the help of 3 pointers & increment the pointers on the basis of some conditions which we will be discussing in the dry run. Also we will be using the set data structure to make sure we have unique elements only.

Dry run

A $\rightarrow \{1, 5, 10, 20, 40, 80\}$ B $\rightarrow \{6, 7, 20, 80, 100\}$

\uparrow_i

\uparrow_j

C $\rightarrow \{3, 4, 15, 20, 30, 70, 80, 120\}$

\uparrow_k

1) $i=0, j=0, k=0$

$A[i] < B[j] \rightarrow i++$

2) $i=1, j=0, k=0$

$A[i] < B[j] \rightarrow i++$

3) $i=2, j=0, k=0$

$A[i] < B[j] \rightarrow \text{false}$

$B[j] < C[k] \rightarrow \text{false}$

hence $k++$

$i^{\circ}=2, j^{\circ}=0, k=1$

$\{1, 5, 10, 20, 40, 80\}$

\uparrow_i

$\{6, 7, 20, 80, 100\}$

\uparrow_j

$\{3, 4, 15, 20, 30, 70, 80, 120\}$

\uparrow_k

4) $10 < 6 \Rightarrow \text{False}$

$6 < 4 \Rightarrow \text{False}$

Hence $k++$

5) $i^{\circ}=2, j^{\circ}=0, k=2$

$A[i] < B[j] \rightarrow \text{False}$

$B[j] < C[k] \rightarrow \text{True and hence } j++$

6) $i^{\circ}=2, j^{\circ}=1, k=2$

$B[j] < C[k] \rightarrow \text{True & hence } j++$

7) $i^{\circ}=2, j^{\circ}=2, k=2$

$\{1, 5, 10, 20, 40, 80\}$

\uparrow_i

$\{6, 7, 20, 80, 100\}$

\uparrow_j

$\{3, 4, 15, 20, 30, 70, 80, 120\}$

\uparrow_k

$A[i] < B[j] \rightarrow \text{True & hence } i++$

8) $i = 3, j = 2, k = 2$

Now $A[i] < B[j] \Rightarrow \text{False}$

$B[j] < C[k] \Rightarrow \text{False}$

Hence simply $k++$

9) $i = 3, j = 2, k = 3$

$\downarrow i$

{1, 5, 10, 20, 40, 80}

$\downarrow j$

{6, 7, 20, 80, 100}

$\downarrow k$

{3, 4, 15, 20, 30, 70, 80, 120}

$A[i] == B[j] \& \& B[j] == C[k]$

and simply add $A[i]$ to the set and do
 $i++, j++$ and $k++$.

Similarly we can find the common elements.

Code

```
void commonElements (vector<int> &A,
vector<int> &B, vector<int> &C) {
```

//Find the size of all the vectors

int n1 = A.size();

int n2 = B.size();

int n3 = C.size();

//make set to store unique elements

set<int>s;

//3 pointer approach

int i = 0;

```
int j = 0;  
int k = 0;
```

//Traverse till any of the array finishes
while ($i < n_1$ & $j < n_2$ & $k < n_3$) {

//Found common element

if ($A[i] == B[j]$ & & $B[j] == C[k]$) {

s.insert (A[i]) ;

i++ ;

i + + *j*

$k++$ } ahead

3

//Sorted array advantage below

```
else if (A[i] < B[j]) {
```

i++;

3

```
else if (B[j] < C[k]) {
```

j + + j

3

else {

R++j

3

3

// Print the common elements

```
for (auto i:s) {
```

```
cout << i << " ";
```

4

3

linear time and space complexity

$$TC = O(n_1 + n_2 + n_3)$$

$$SC = O(n_1 + n_2 + n_3)$$