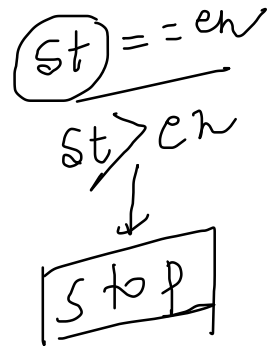
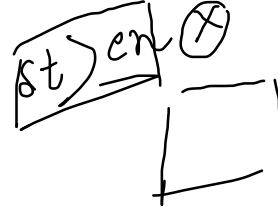
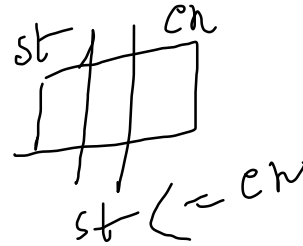
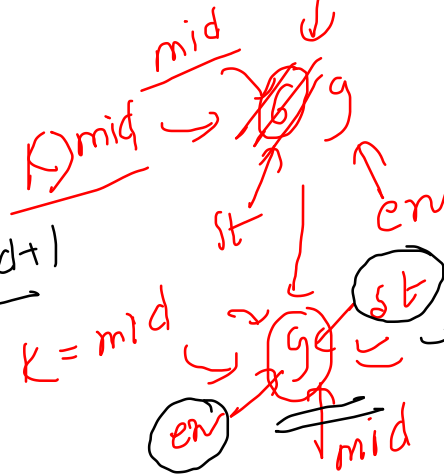
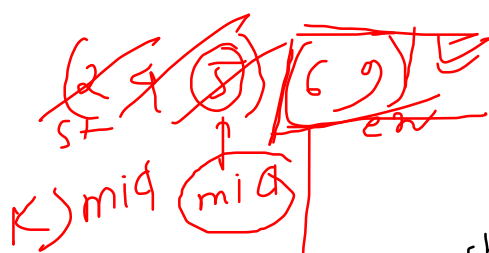
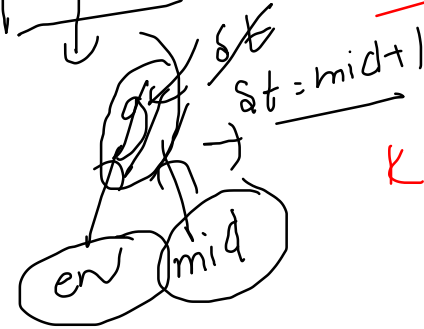
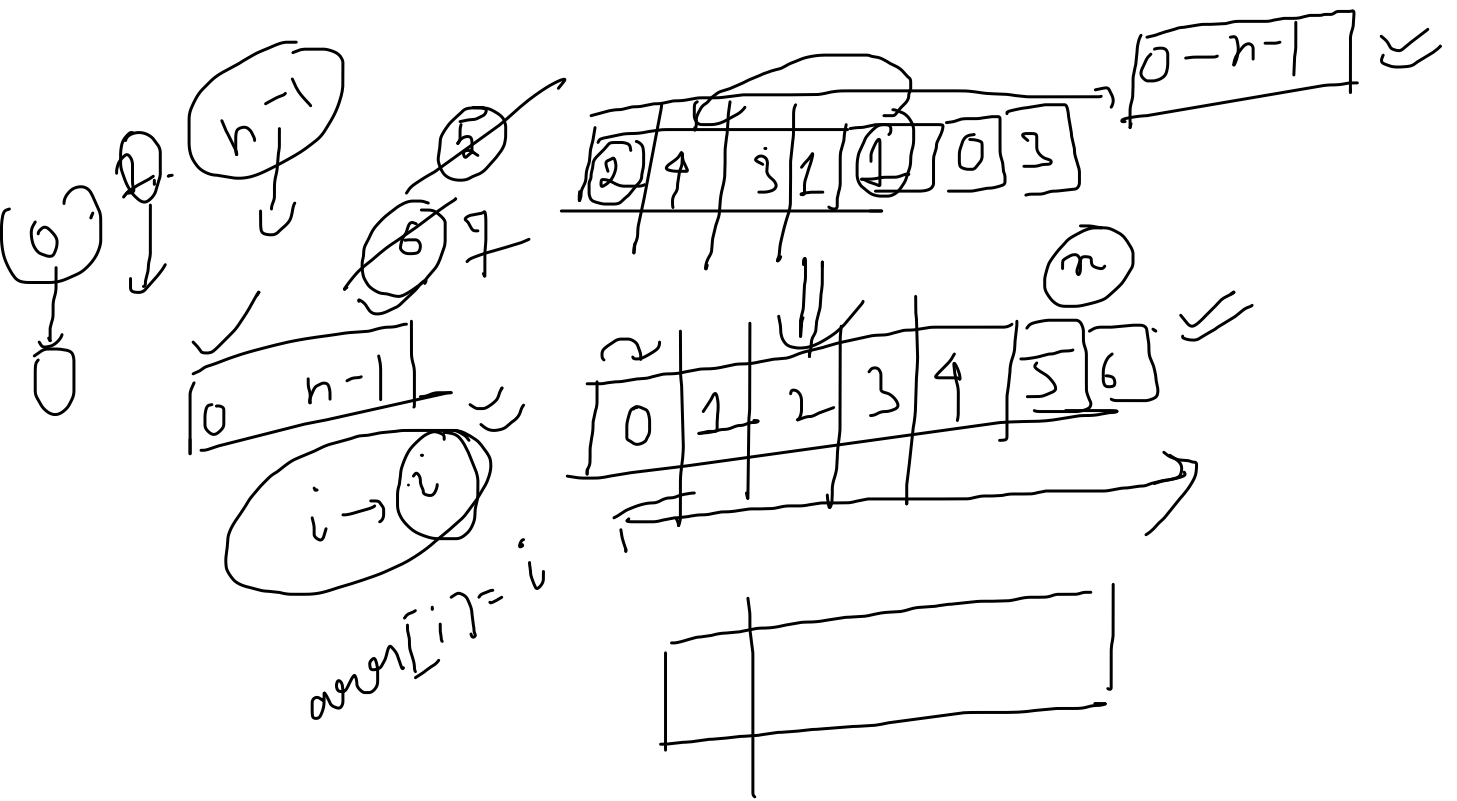


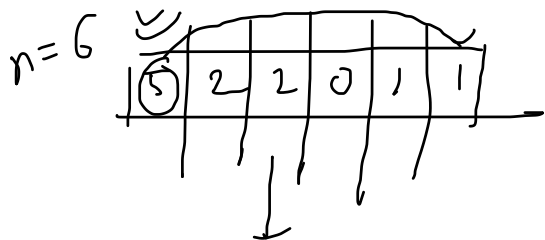
mid=4
 $st = 4$
 $en = 4$

$k = 9$

mid < K

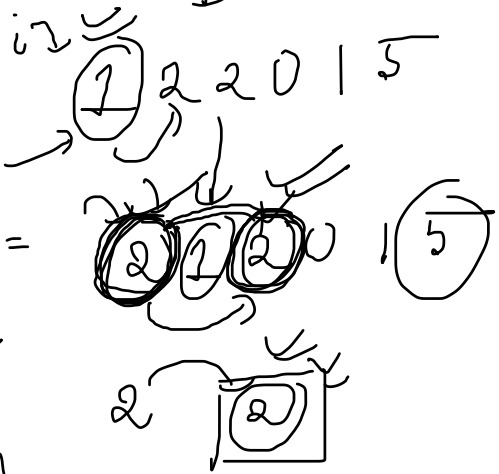






$arr[i] \neq i$

$arr[i] \neq i$
 $while(arr[i] \neq i)$

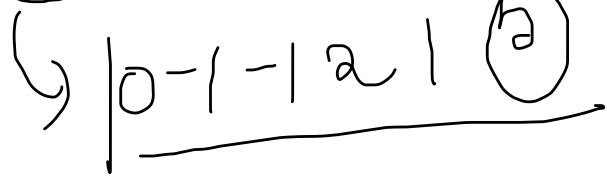
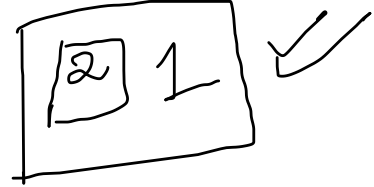
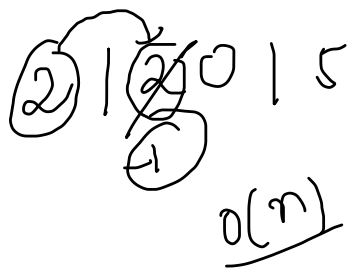


$arr[i]$

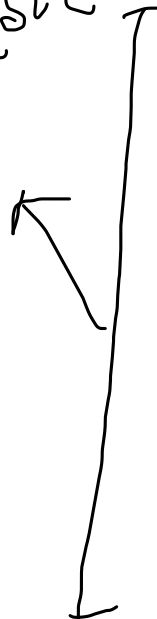


2

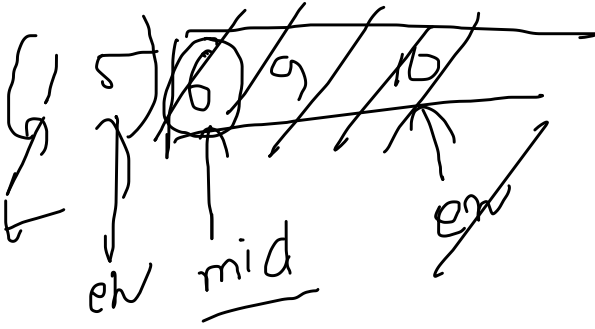




```
binarySearch(arr, st, en) {  
    while(st <= en) {  
        mid = (st + en) / 2;  
        if (arr[mid] == k) {  
            print(mid);  
            break;  
        }  
        else if (arr[mid] > k) {  
            en = mid - 1;  
        }  
        else {  
            st = mid + 1;  
        }  
    }  
}
```



$$k = 1$$



$$mid, (k)$$

$$mid > k$$

$$mid > k$$

$$mid - 1$$

st

$$mid == k$$



$$5$$

$$en$$

en

$$k > \frac{mid}{mid + 1}$$



$\frac{n}{2^0}, \frac{n}{2^1}, \frac{n}{2^2} \dots \frac{n}{2^x}$

$(x+1)$

$\log_2(n)$

(1)

$x = \log_2(n)$

$\frac{n}{2^x} = (1) \Rightarrow$

$2^x = n$

$x \cdot \log_2(2) = \log_2(n)$

worst case

$(n) \rightarrow 1$

$n/2 \rightarrow 1$

$n/4 \rightarrow 1$

$n/8 \rightarrow 1$

$(1) \rightarrow 1$

$\frac{n}{2^0}$

$\frac{n}{2^1}$

$\frac{n}{2^2}$

$\frac{n}{2^3}$

\vdots

$(x+1)$

$\log_2 n + 1$

$O(\log_2 n + 1)$

$O(\log n)$

$$\left(\frac{n}{2^0}\right), \frac{n}{2^1}, \left(\frac{n}{2^2}\right), \left(\frac{n}{2^3}\right) \dots \frac{n}{2^x} = 1$$

$$n \rightarrow 1$$

$$\frac{n}{2} \rightarrow 1$$

$$\frac{n}{4} \rightarrow 1$$

$$\boxed{0} \left(\log_2 n \right)$$

$$\boxed{O(\log n)}$$

$$2^0 = 1$$

$$\frac{n}{2^0} = n$$

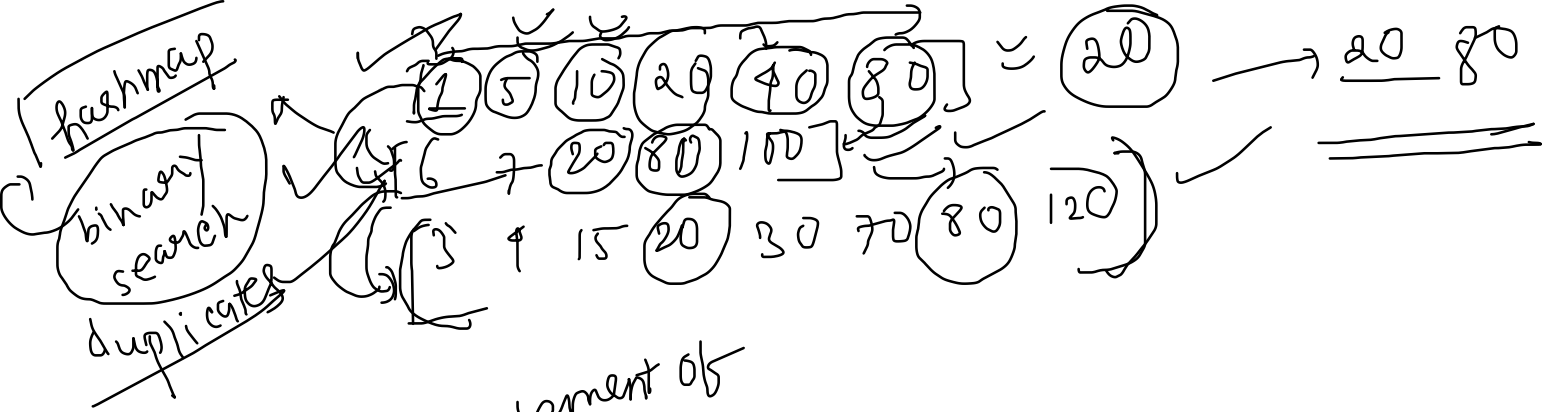
$$\frac{n}{2^x} = 1 \rightarrow \frac{2^x}{n} = 1$$

$$2^x = n$$

$$\log_2(n) = 1$$

$$x \log_2(2) = \log_2(n)$$

$$x \cdot 2 = \log_2(n) \Rightarrow x = \log_2(n)$$

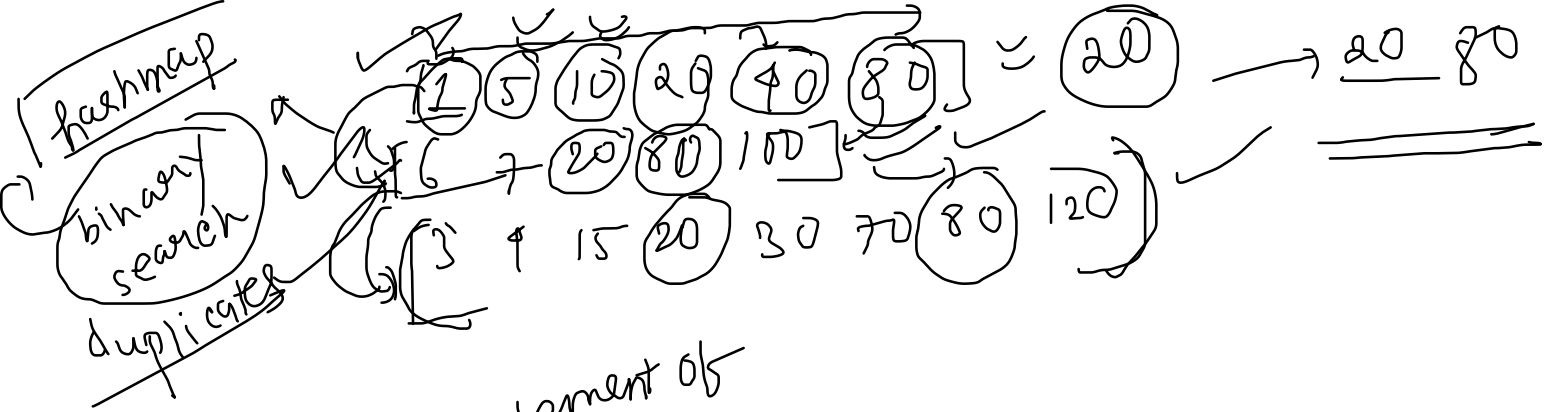


for each element of
array 1

search in array 2
if present

array 3

if present
print as final
answer

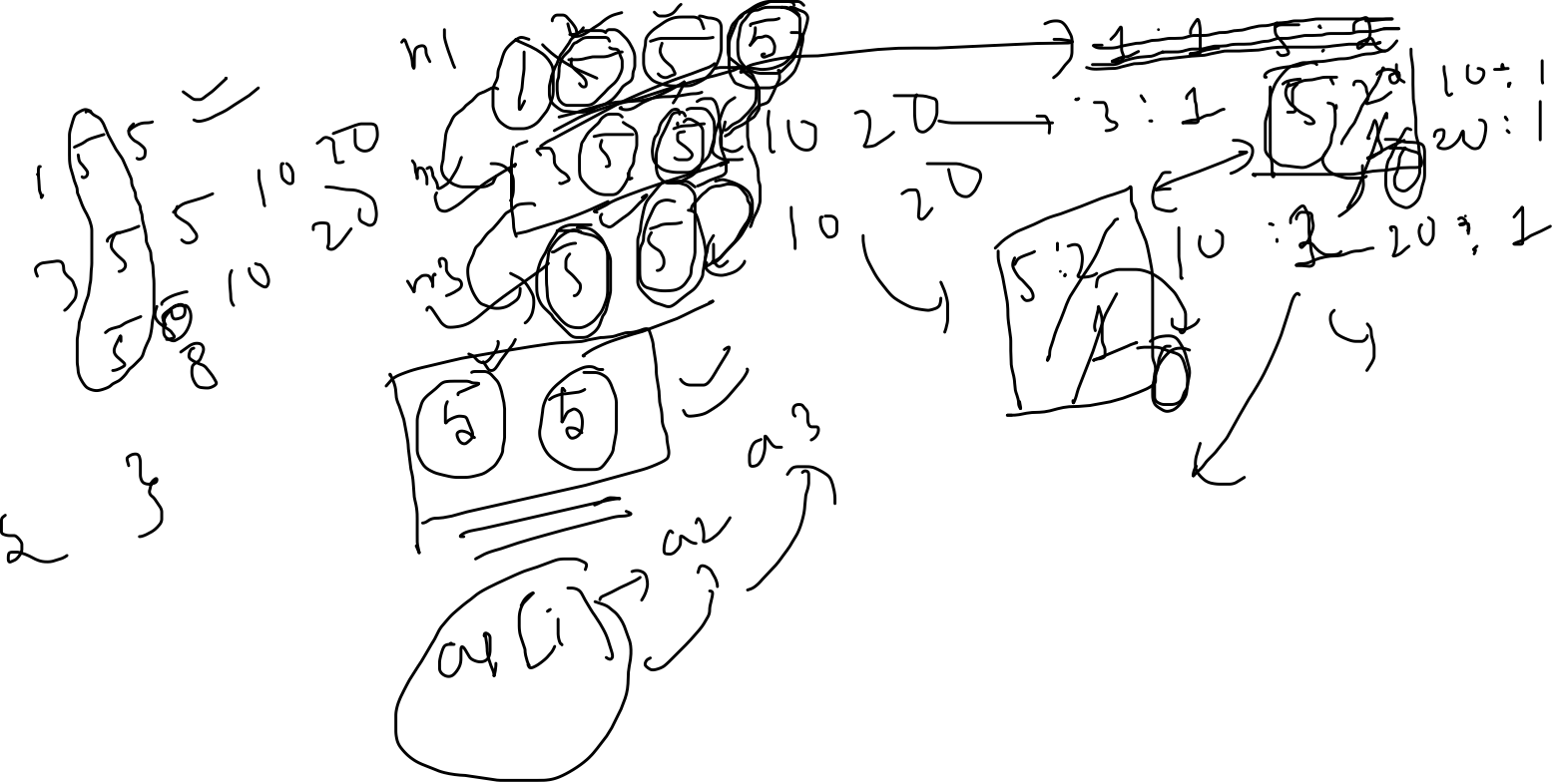


for each element of
array 1

↳ search in array 2
↳ if present

↳ array 3

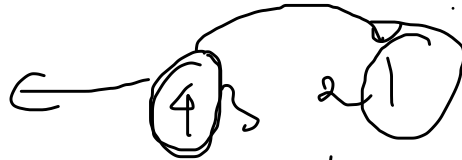
↳ if present
print as final
answer



4 → ③



① 2 3 4



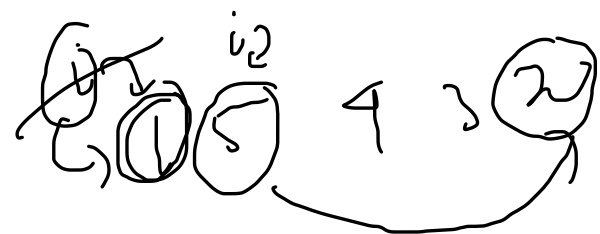
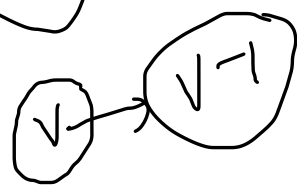
~~arr~~ 4 → arr[4-1]



3 → ②

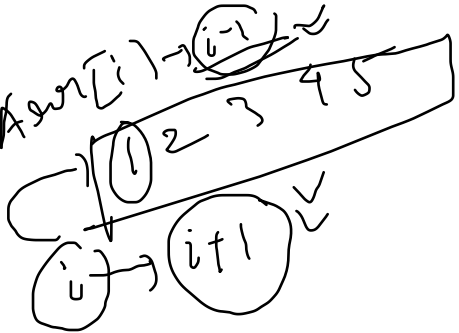
arr[3-1]

1 2 3 4



cycle
sort





cycle sort

