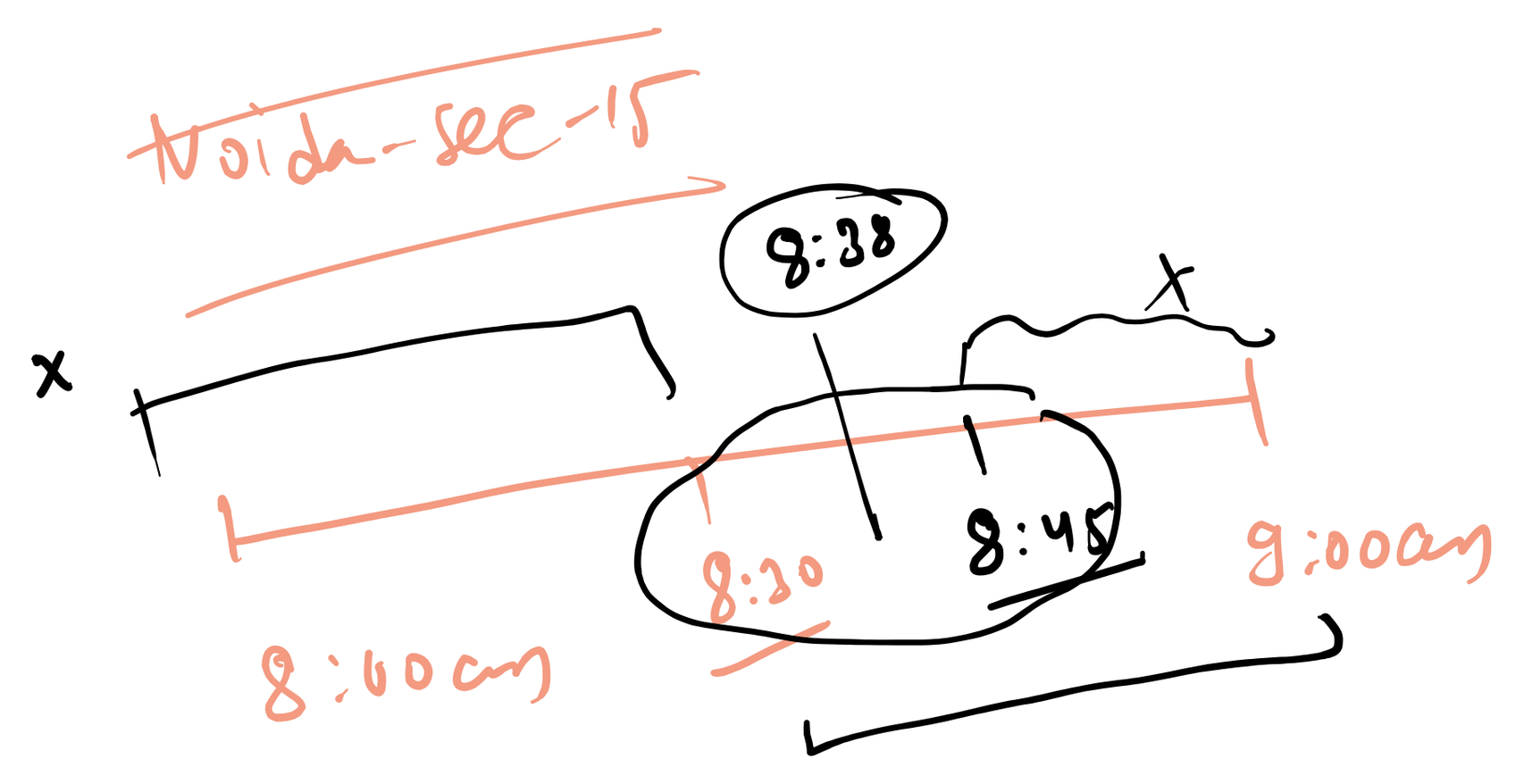
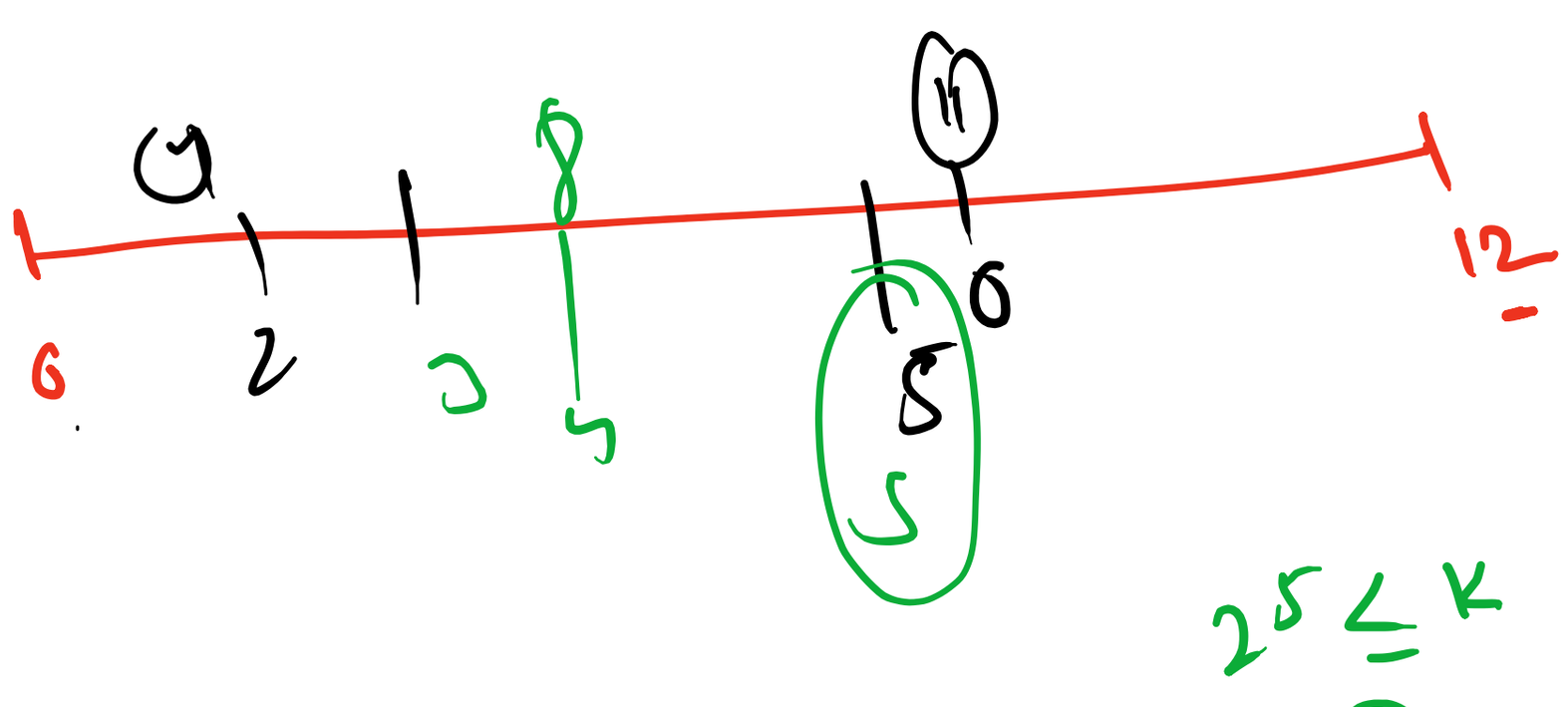
Lec-10

29/12/24 4:40 PM

Noida-sec-15



[2, 3, 4, 7, 8, 9, 11, 13, 17, 19, 20, 23, 27]   (10)
0 1 2 3 4 5 6 7 8 9 10 11 12



while (lo<=hi) {
mid = (lo+hi)/2
if(arr[mid] == item) {
ret True
}
else if(arr[mid]>item){
hi=mid-1
}
else {
lo=mid+1
}
}

$x^k <= n$

$k <= \log_x n$

$F(n) = x^k \le n$

$2^5 \le k$

$x^{k} \le n$

$k <= \log_{10} x^n$

x -> maximum

n=147

k=3

5 ✓

x=1

while(x^k<=n){

x++

}

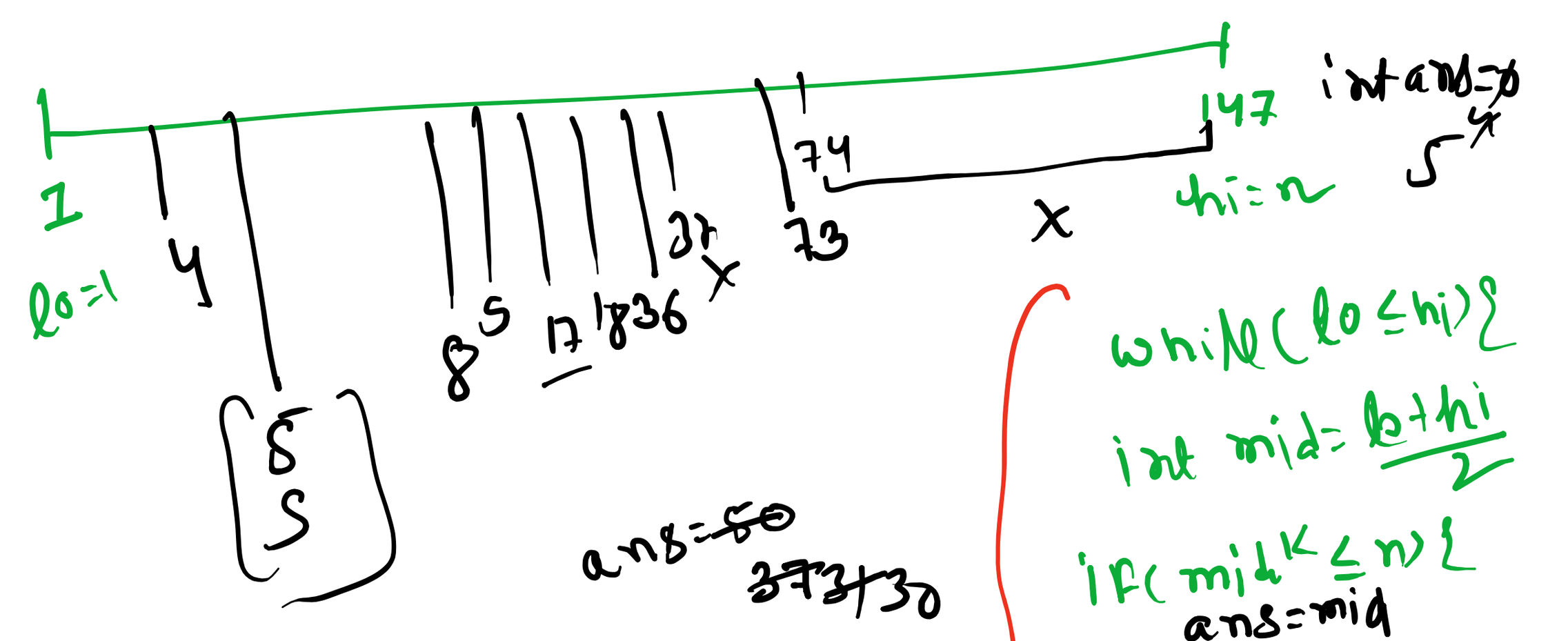
ret (x-1)

$1^3 <= 147$
$2^3 <= 147$
$3^3 <= 147$
$4^3 <= 147$
$5^3 <= 147$
6^3

k=3

N=147

$7^3 \le N$

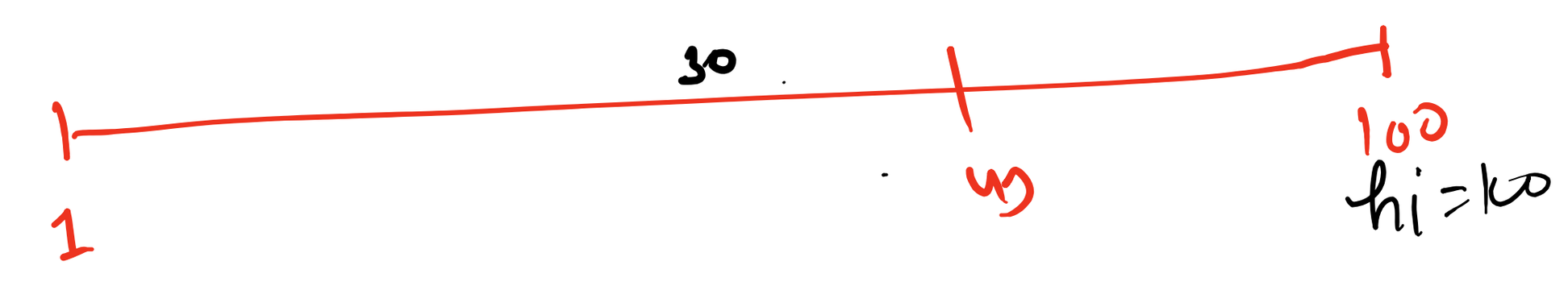


int ans=0

ans = ~~50~~ 37 ~~37~~ 30

while(lo<=hi){
int mid=(lo+hi)/2
if(mid^k<=n){
ans=mid
lo=mid+1
}
else
hi=mid-1
}

N=100   bad=30



lo=1

while(lo<=hi) {
Int mid = (lo+hi)/2
if(isbadVersion(mid)){
ans = mid
hi=mid-1
}
else{
lo=mid+1
}
}

```
public static int firstBadVersion(int n) {
        int lo = 1;
        int hi = n;
        int ans = 0;
        while (lo <= hi) {
            int mid = (lo + hi) / 2;
            if (isBadVersion(mid)) {
                ans = mid;
                hi = mid - 1;
            } else {
                lo = mid + 1;
            }
        }
        return ans;
    }
```

$\frac{lo+hi}{2} = lo + \frac{(hi-lo)}{2}$

$= \frac{2 \times lo + hi - lo}{2}$

$= \frac{hi+lo}{2}$

$n = 2^{31}-1$ bad $= 2^{31}-3$

$lo = 2^{31}-6$ $hi = 2^{31}-1$

$mid = \frac{2^{31}-6+2^{31}-1}{2}$

$= \frac{2 \times 2^{31}-7}{2} = \frac{2^{32}-7}{2} = 2^{31}-3$

$2^{31}-6+\frac{(2^{31}-1-2^{31}+6)}{2}$

$2^{31}-6+\frac{5}{2}$

$2^{31}-6+2 = 2^{31}-4$

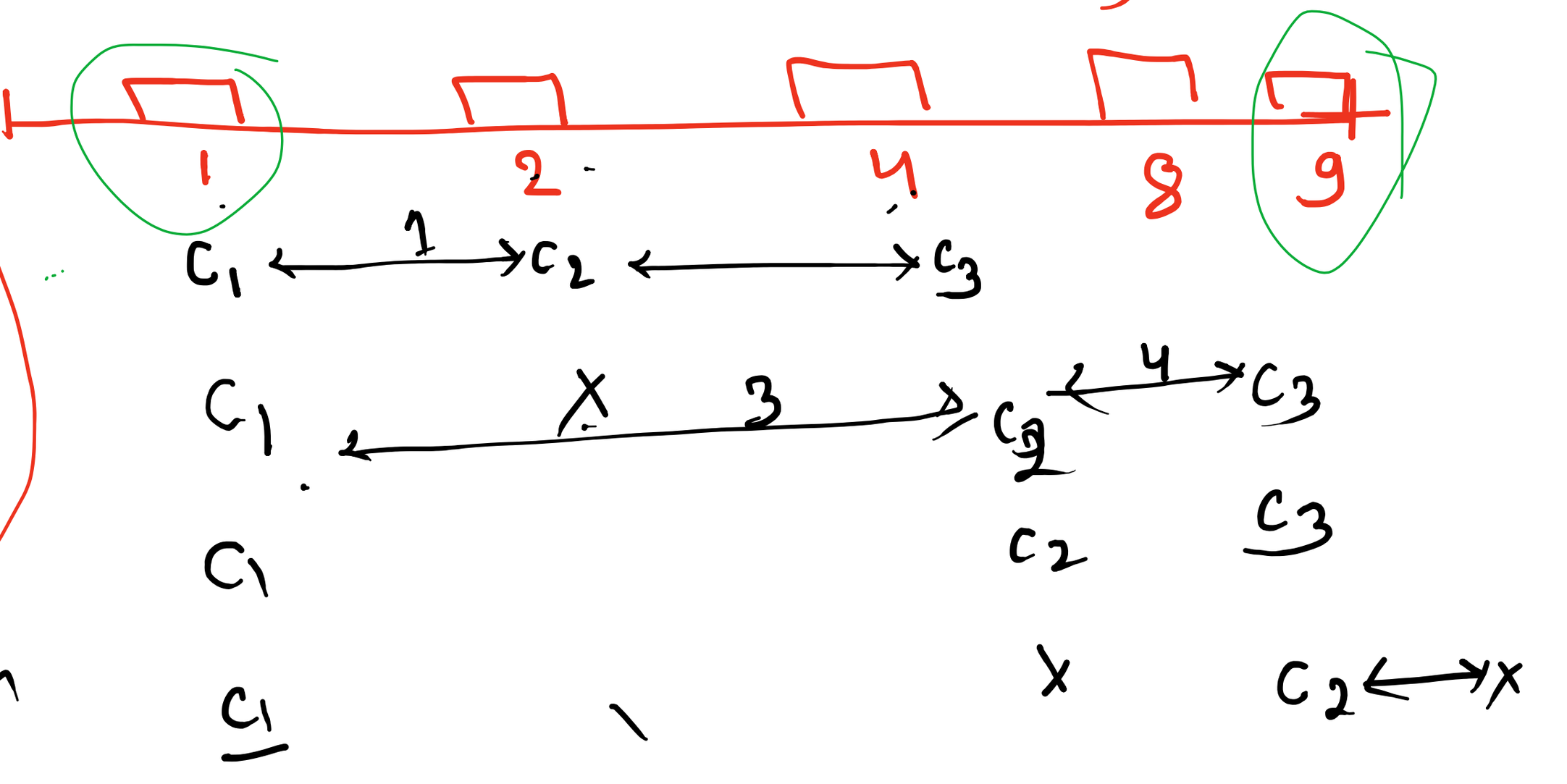
1
5 3
1
2
8
4
9

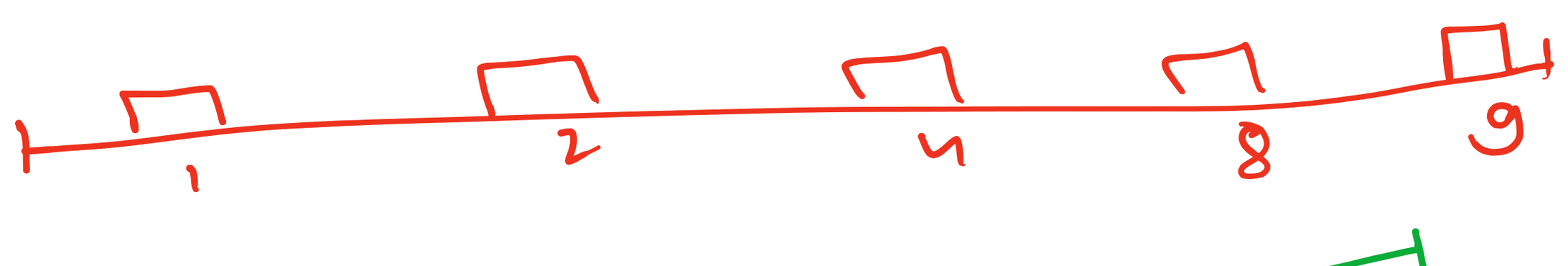
Sol

ans = 3

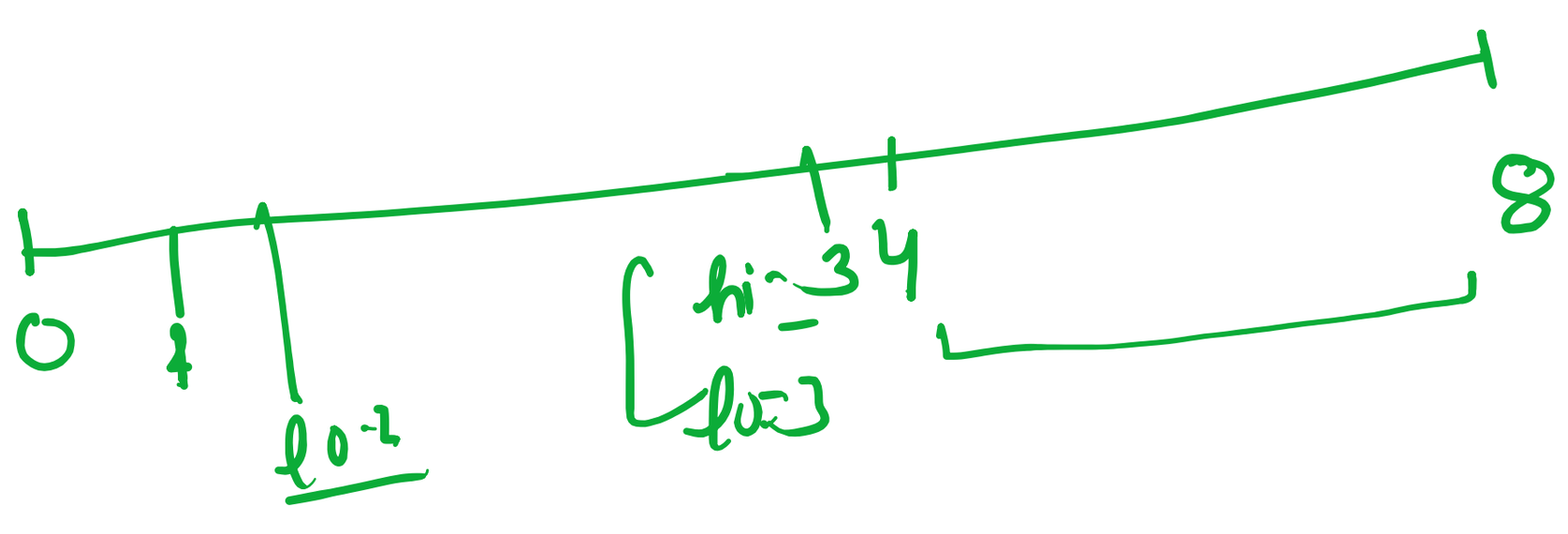
min=1
min=2
min=3
min=4

x=1 while( ){ x++ } x-1





while(lo<=hi){
mid=(lo+hi)/2
if(isitposs( )){
ans=mid
lo=mid+1
}
else
hi=mid-1
}
}



C(1, 2, 4, 8, 9)

```
public static int Mini_Dis(int[] stall, int noc) {
    int lo = 0;
    int hi = stall[stall.length - 1] - stall[0];// 9-1=8
    int ans = 0;
    while (lo <= hi) {
        int mid = (lo + hi) / 2;
        if (isitpossible(stall, noc, mid)) {
            ans = mid;
            lo = mid + 1;
        } else {
            hi = mid - 1;
        }
    }
    return ans;
}
public static boolean isitpossible(int[] stall, int noc, int mid) {
    // TODO Auto-generated method stub
    int cow=1, pos = stall[0]
    For(int i=1  i<stall.len  i++) {
        if( stall[i] - pos >= mid) {
            pos = stall[i]  cow++
        }
    }
    if(cow == noc)
        return True
}
```