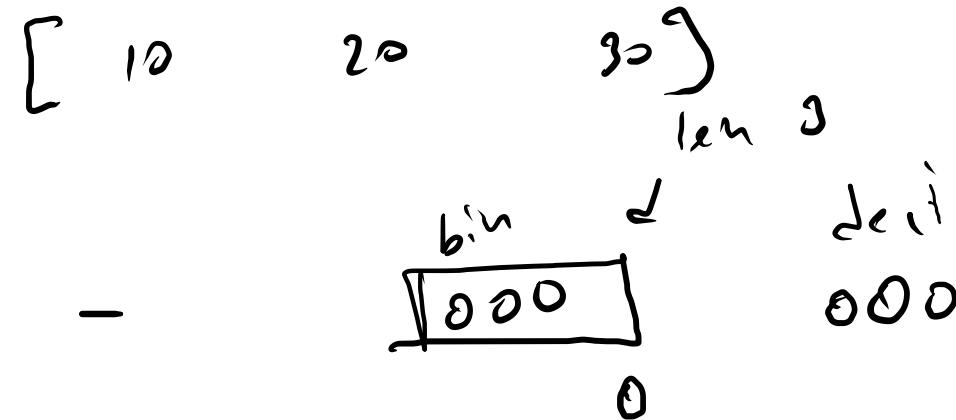


$n=3$



$n/2 = 3$

$$\text{bin} = \begin{matrix} 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow \\ 10 & 20 & 30 \end{matrix}$$

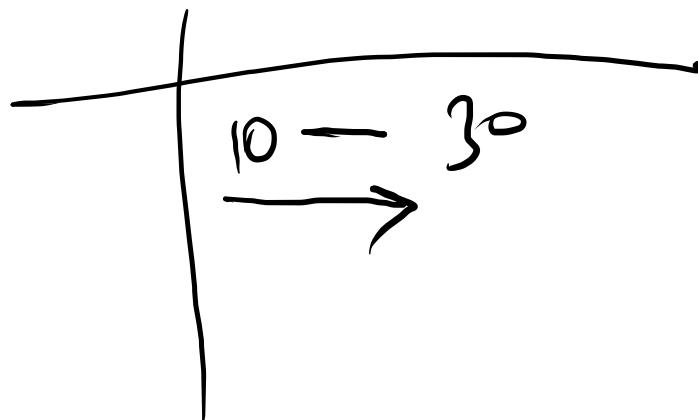
→

10      -      30

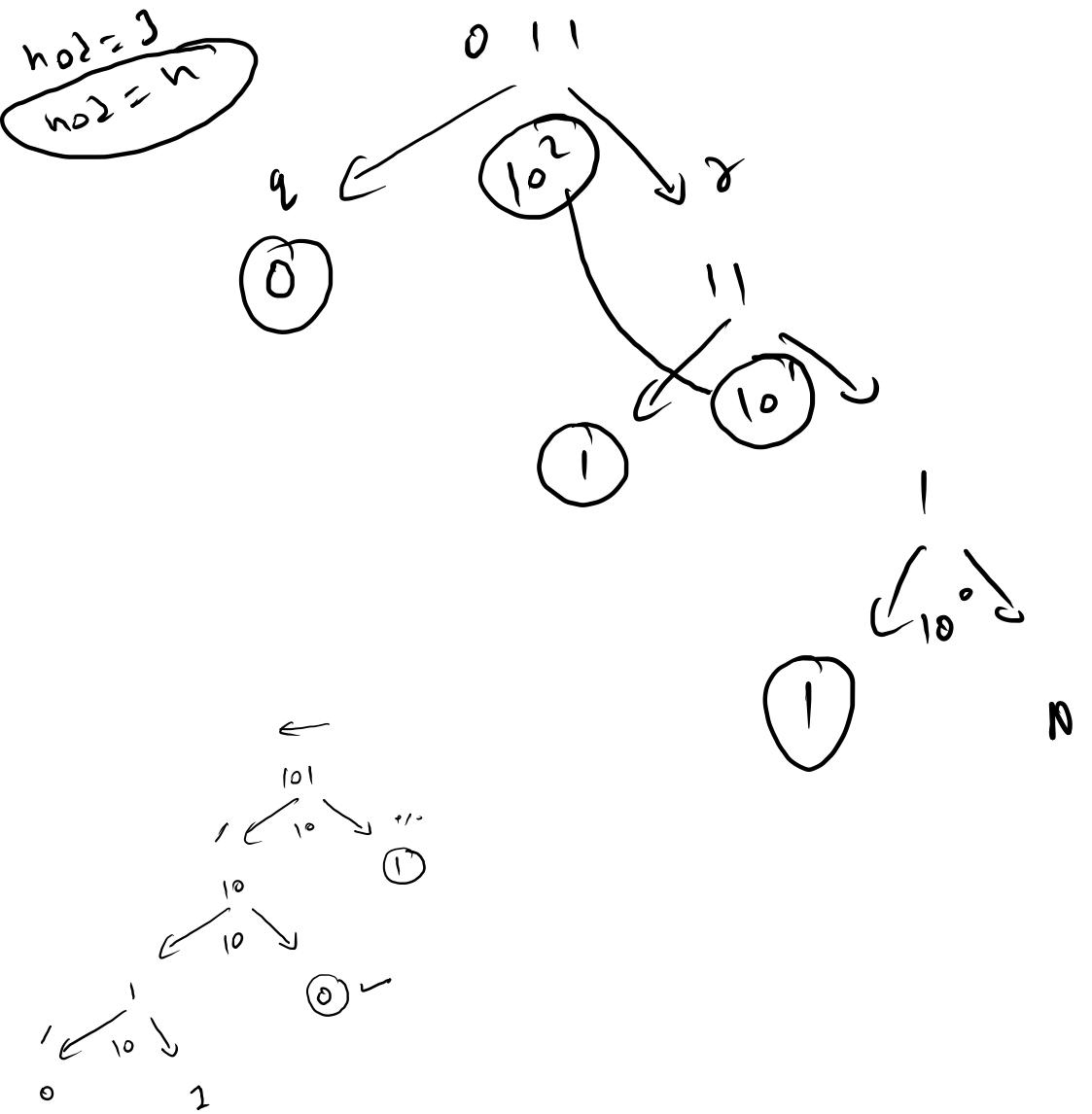
-      20      30

011

$b/2 = 0/1$



if ( $b/2 == 1$ )  
print (arr[i]);  
else  
print ("-");



deci

0 do  $2^n - 1$   
bin = deci2bin(deci)

div =  $10^{n-1}$

for( int i=0; i < n; i++ ) {

q = bin / div;

r = bin % div;

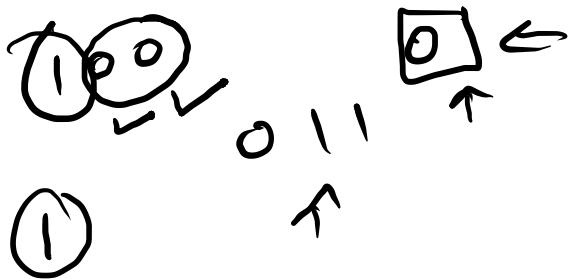
if ( q == 0 )  
-

else arr[i]

bin = r

div = div / 10;

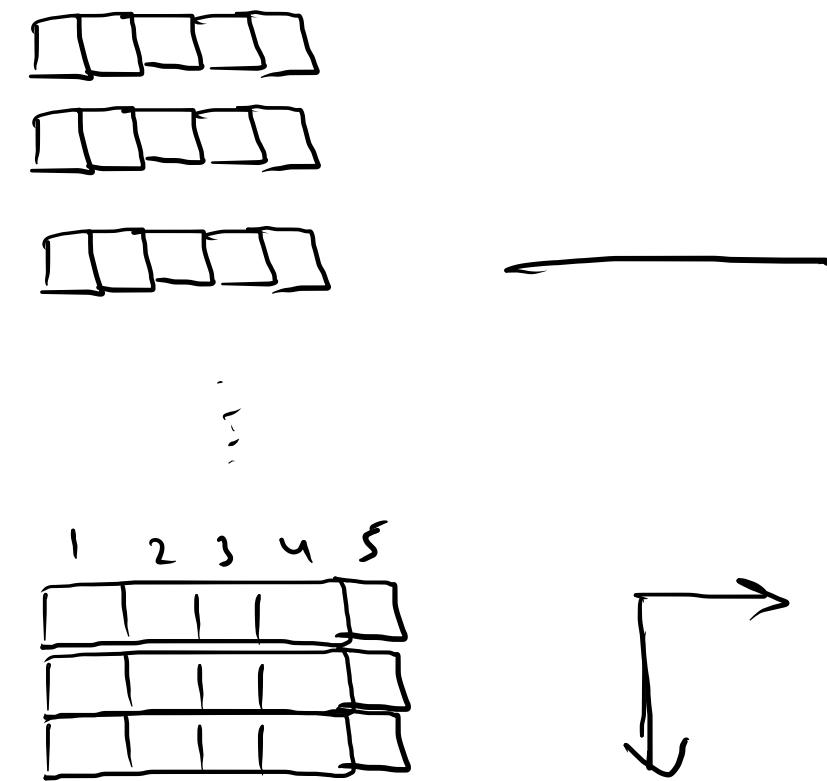
}



while( bin > 0 )

```

int mark = 75
int marks[] = new int[5];
int marks2[] = _____;
int marks3[] = _____;
int marks3[][] = new int[3][5]
    
```



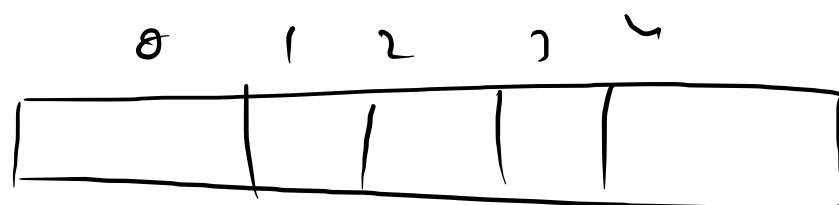
`int matrix [ ] [ ] = new int [ 3 ] [ 5 ];`

	0	1	2	3	4
0					
1			20		
2					

j

	0	1	2	3	4
0	0,0	0,1	0,2		
1		1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4

i



matrix [ i ] [ j ]  
 ↑  
 row      ↑  
 col

$\text{matrix}[1][2] = 20$

② 4 11 12  
 ✓ 13 14 21  
 22 23 24

$n = 2$   
 $m = 4$   
 nxm input  


	0	1	2	3
0	11	12	13	14
1	21	22	23	24

int marks[3][4] = new int[3][4],  
 marks[i][j] = so;

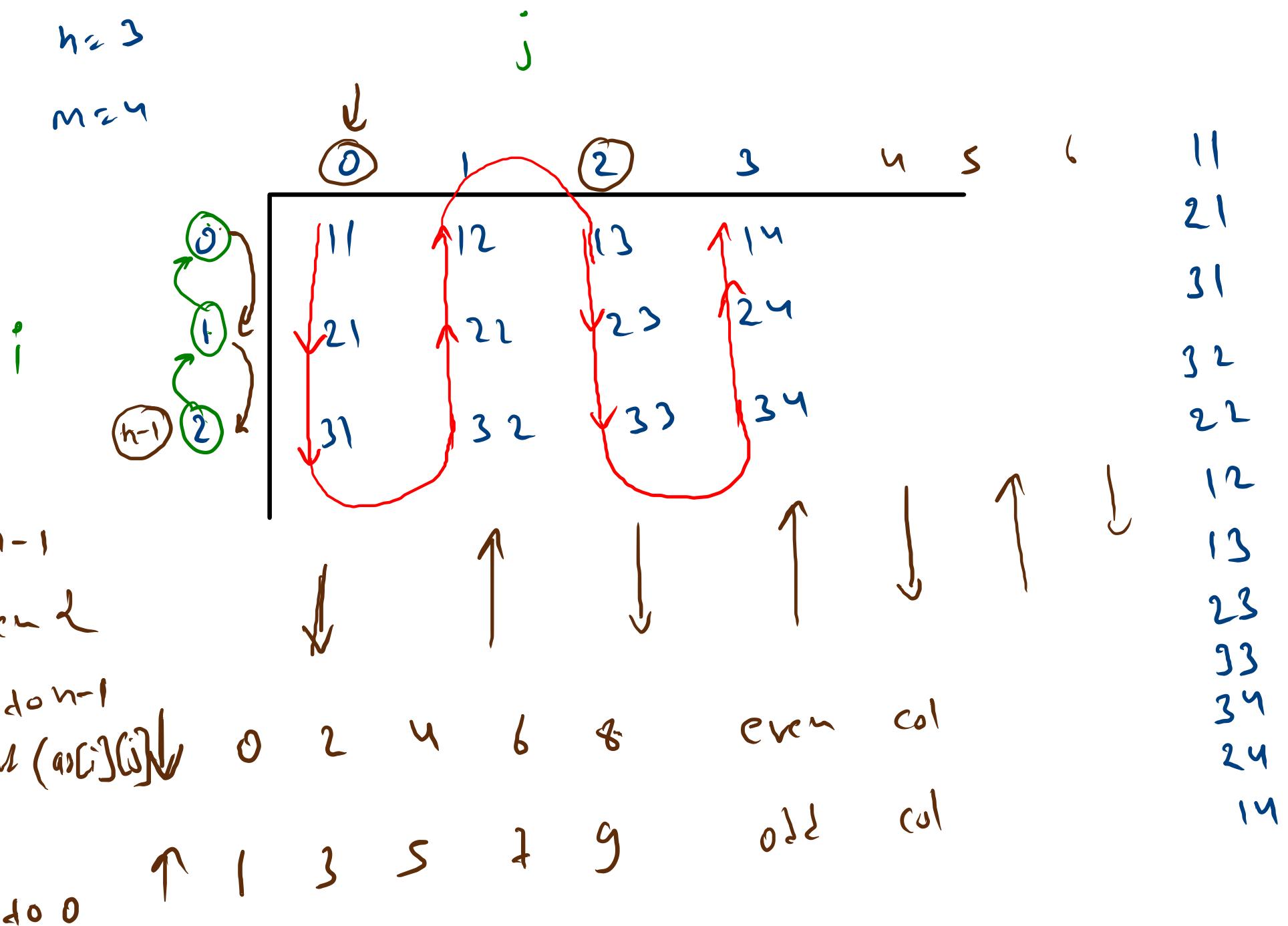
	0	1	2	3	4
0	0,0	0,1	0,2		
1		1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4

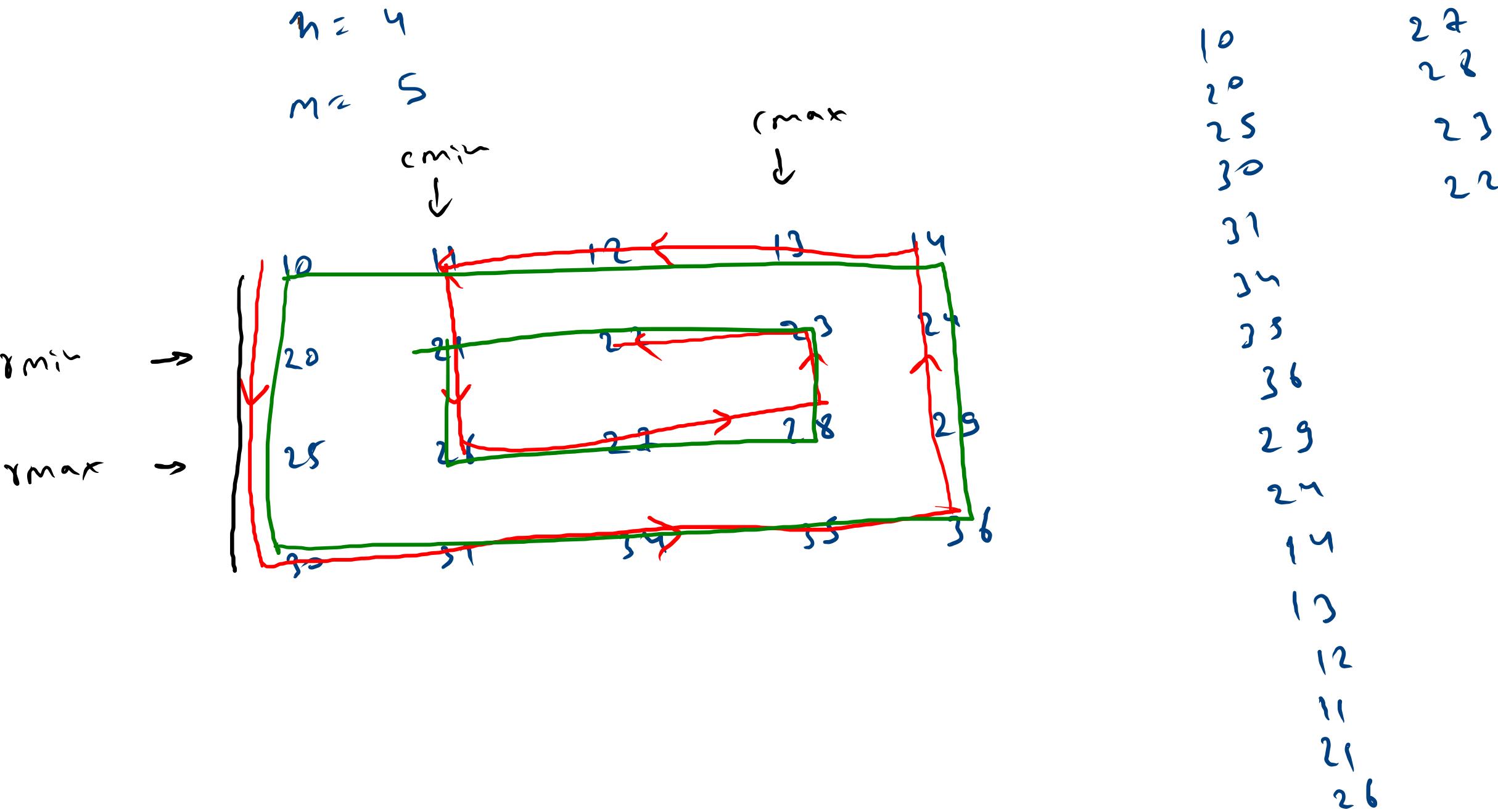
11 12 13 14  
 21 22 23 24

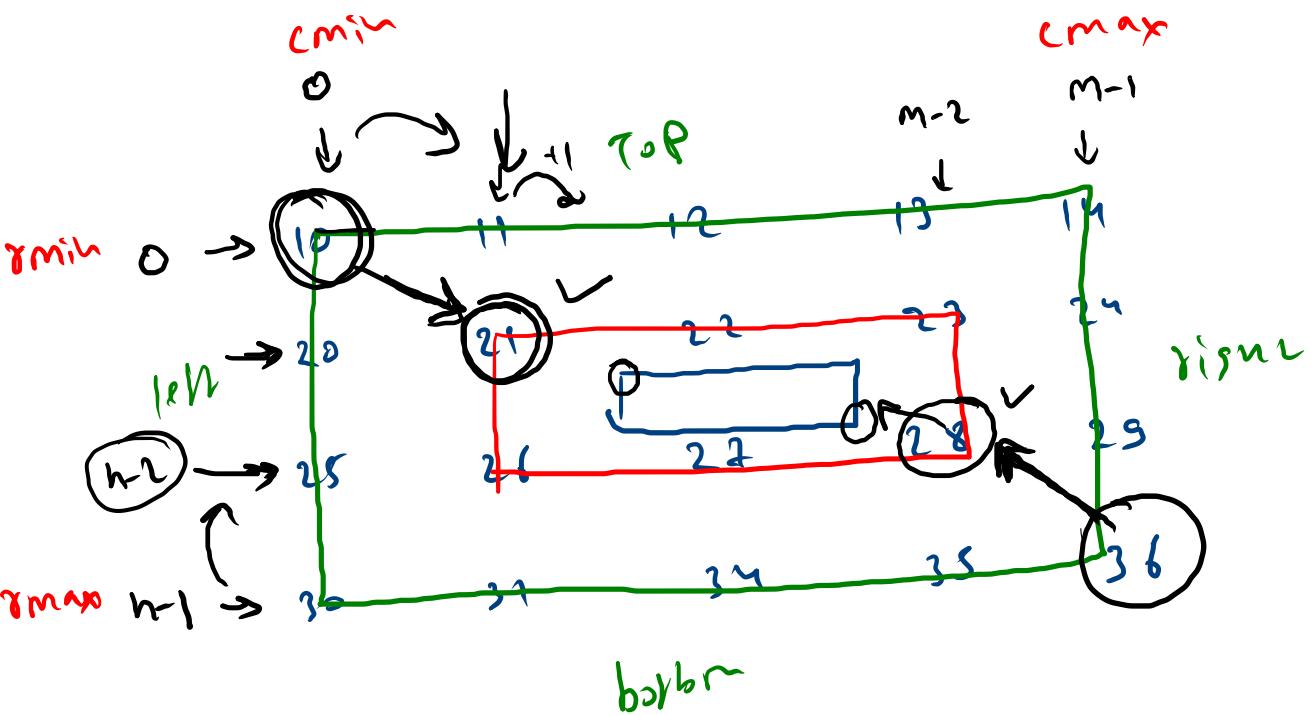
0,0
0,1
0,2
0,3
1,0
1,1
1,2
1,3

h = 3

M-24







```

while (    ) {
    prim left wall
    prim bottom wall
    right
    TOP
}
  
```

change rectangle

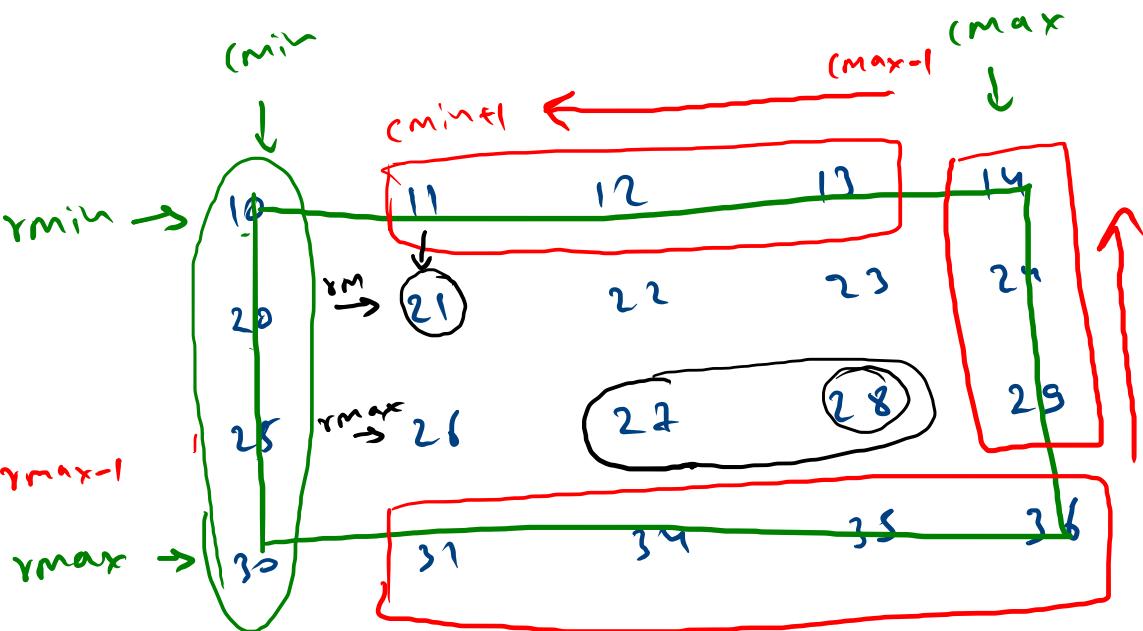
first rectangle

$r_{min}$	$c_{min}$	$r_{max}$	$c_{max}$
0	0	$n-1$	$m-1$
1	1	$n-2$	$m-2$

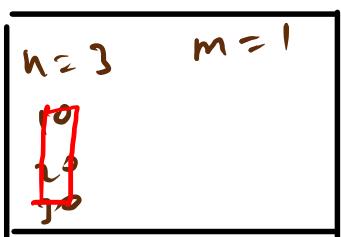
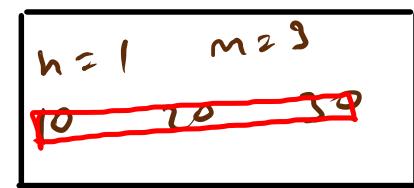
second run

$y_{min}$  ~

$r_{min}++$   
 $c_{min}++$   
 $r_{max}--$   
 $c_{max}--$



$$\text{Total} = h \times m$$



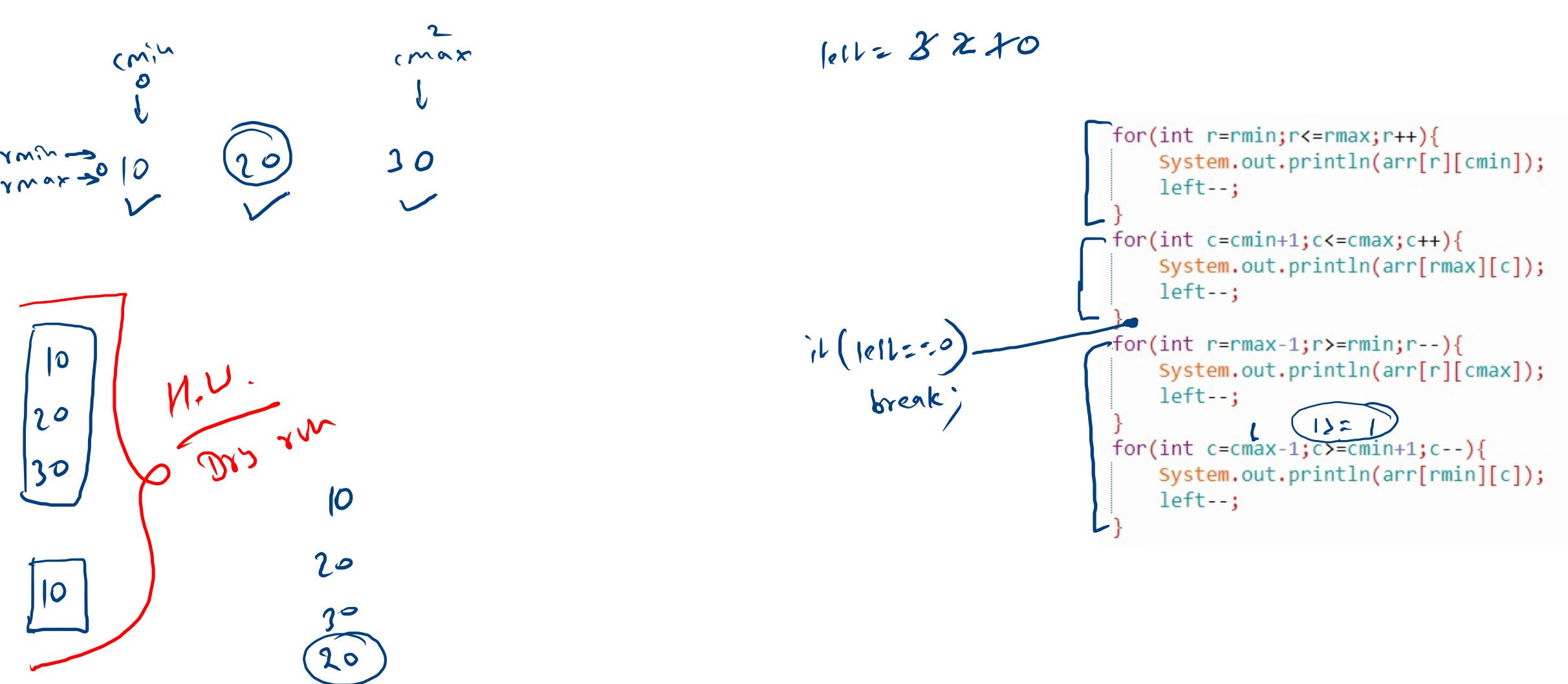
```
int left = hxm
rmin=0; cmin=0
rmax=n-1; cmax=M-1
while (left > 0) {
    for (int r=rmin; r<=rmax; r++)
        arr[r][cmin] // left --
    left //
}
```

```
bottom // for (int c=cmin; c<=cmax; c++)
arr[rmax][c]; left --
```

```
right // r < rmax-1; r>=rmin; r-
arr[r][cmax]; left --
```

```
top // c < cmax-1; c>=cmin; c-
arr[rmin][c]; left --
```

```
(char) // rmin++; cmin++; rmax--; cmax--;
```



int arr[] = new int [4];

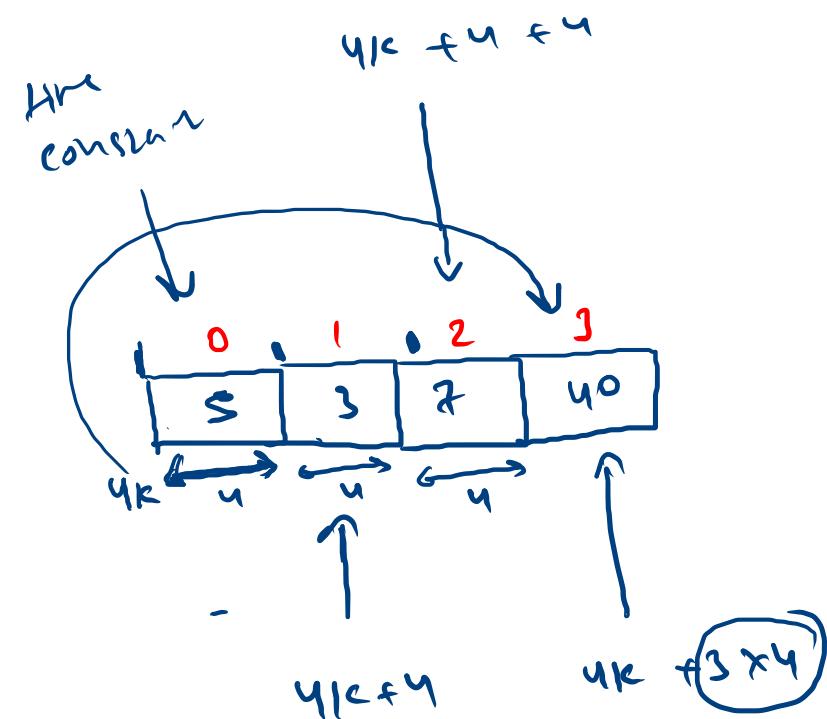
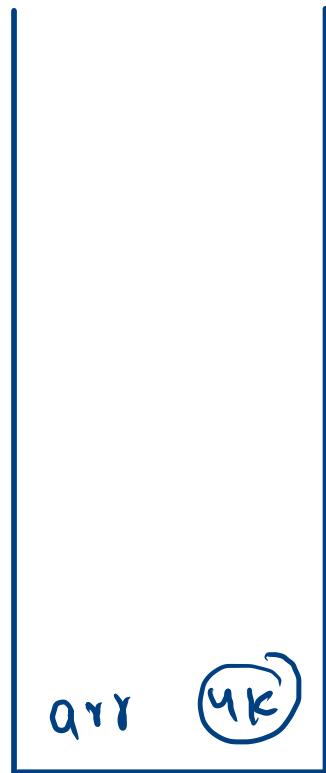
size  
int → 4 byte

arr[i] = 40

4K + i × 4

arr[0]

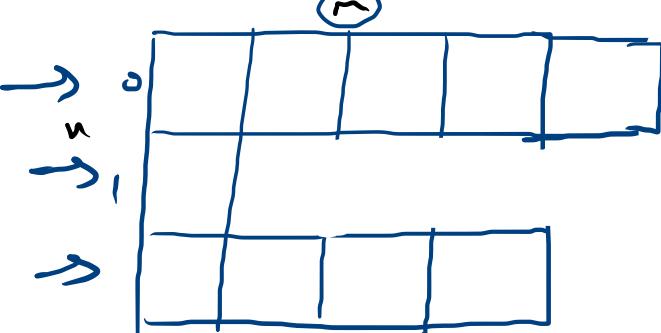
arr[2]



int arr[3] = new int[3][4];  
 int a[3] = new int[5];

~~arr[0] = a;~~

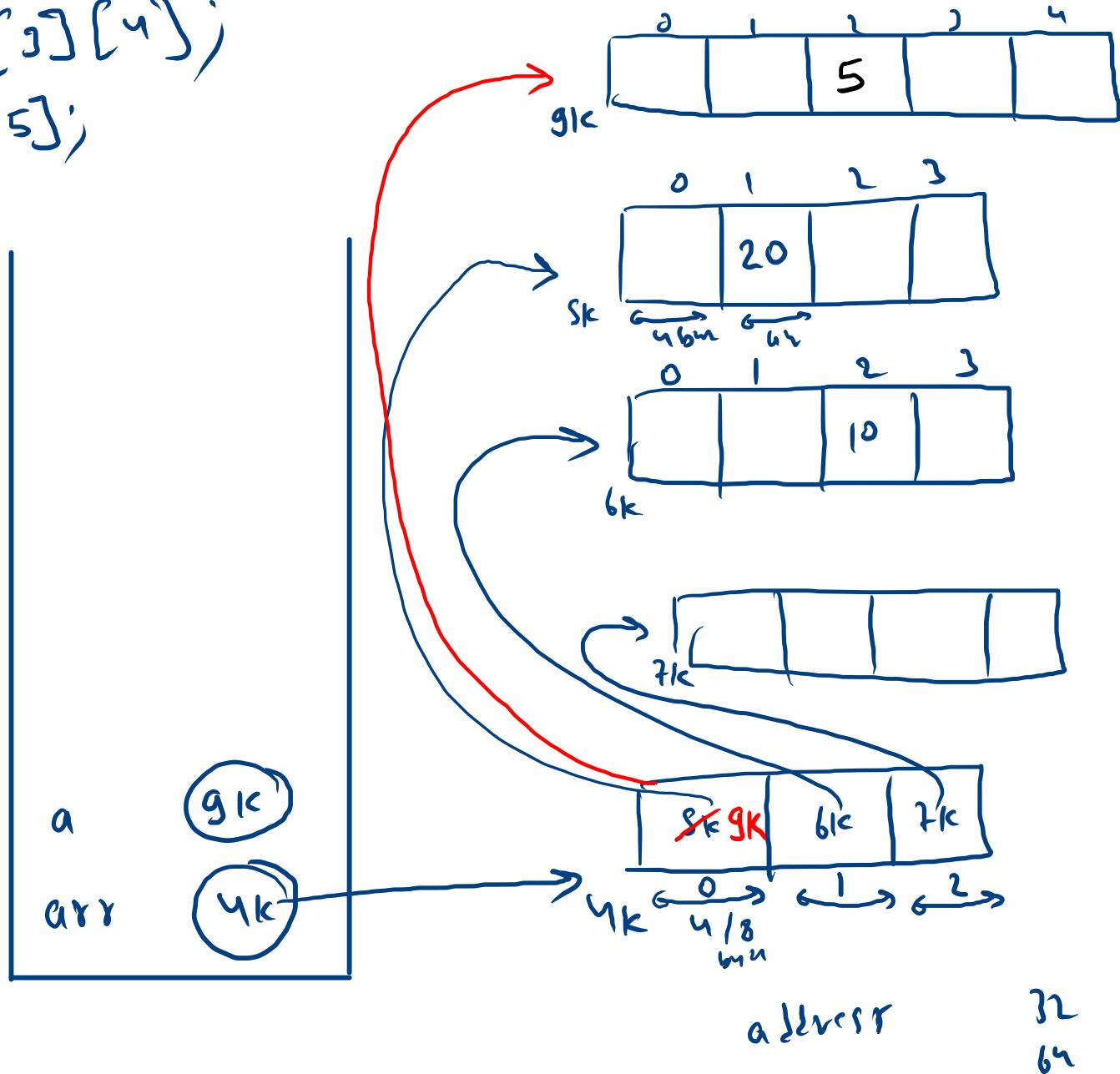
arr[0][2] = 5;



arr[1][2] = 10

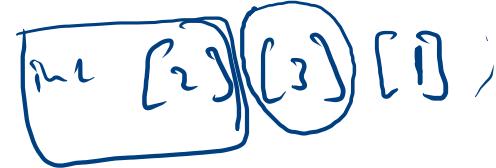
arr[0][1] = 20

②

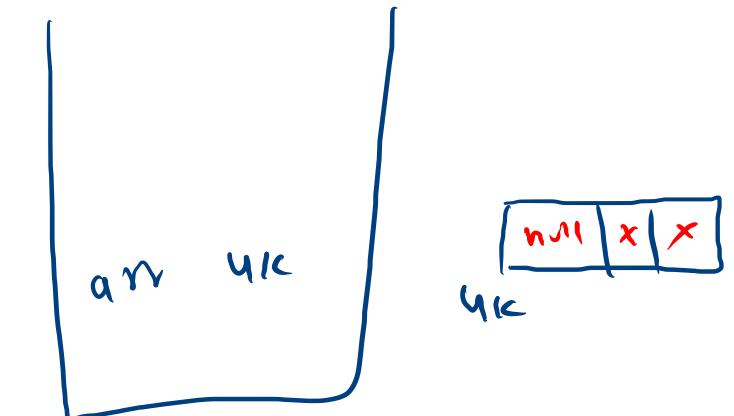
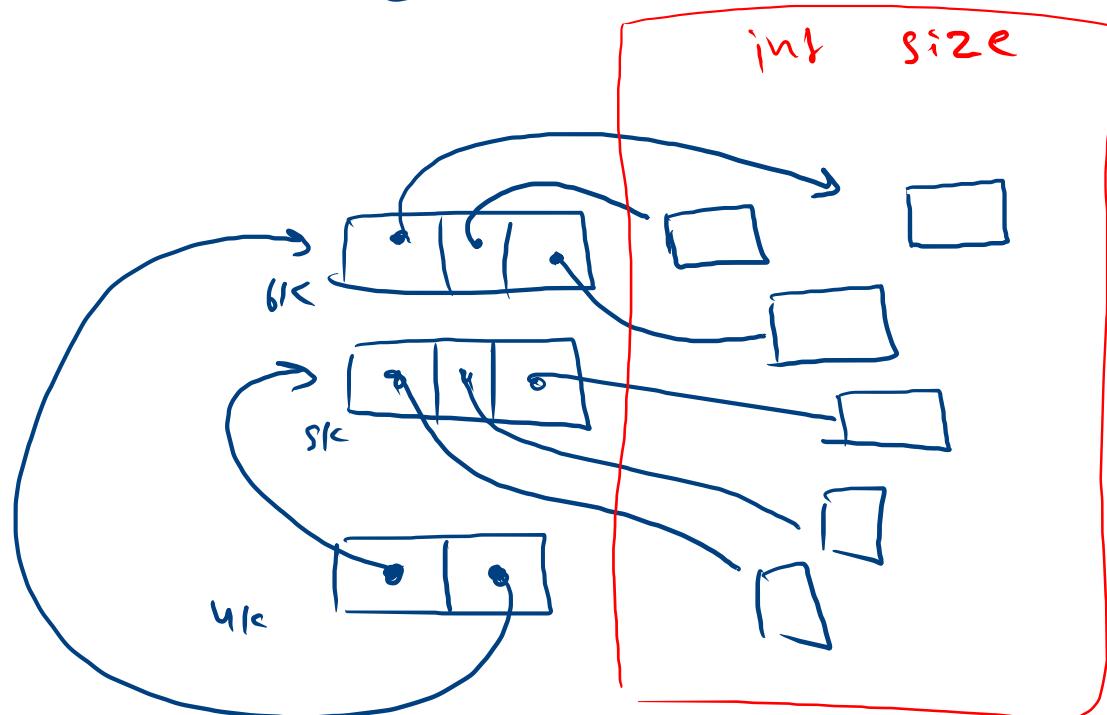
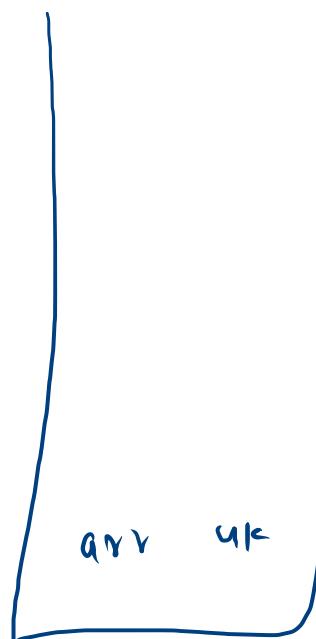


int

arr[ ] [ ] [ ] = new



✓ int arr[ ] [ ] = new int [2] [3];



✗ int arr[ ] [ ] = new int [ ] [3];

