n = 5

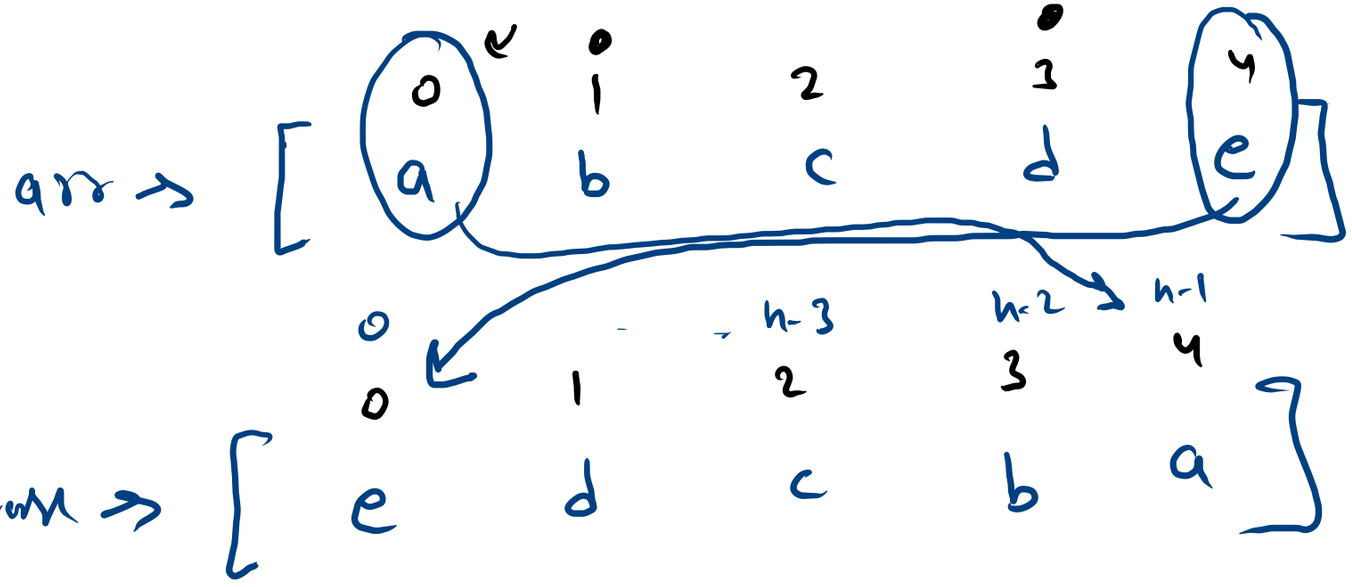
arr → [3 4 2 5 1]

0	1	2	3	4
3	4	2	5	1

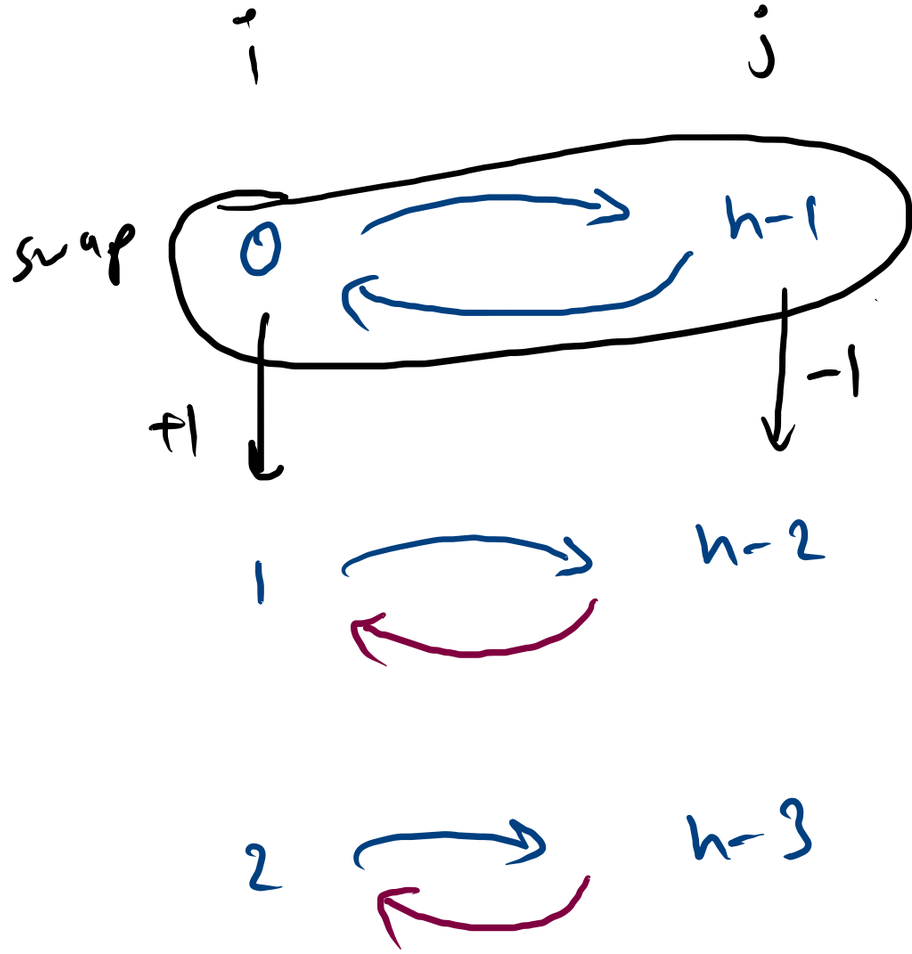
{
3
4
2
5
1
}

arr → [1 5 2 4 3]

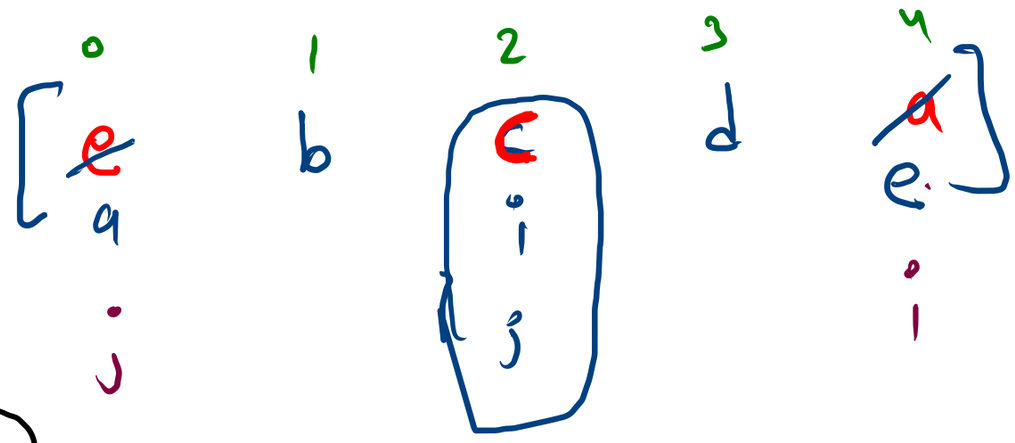
0	1	2	3	4
1	5	2	4	3

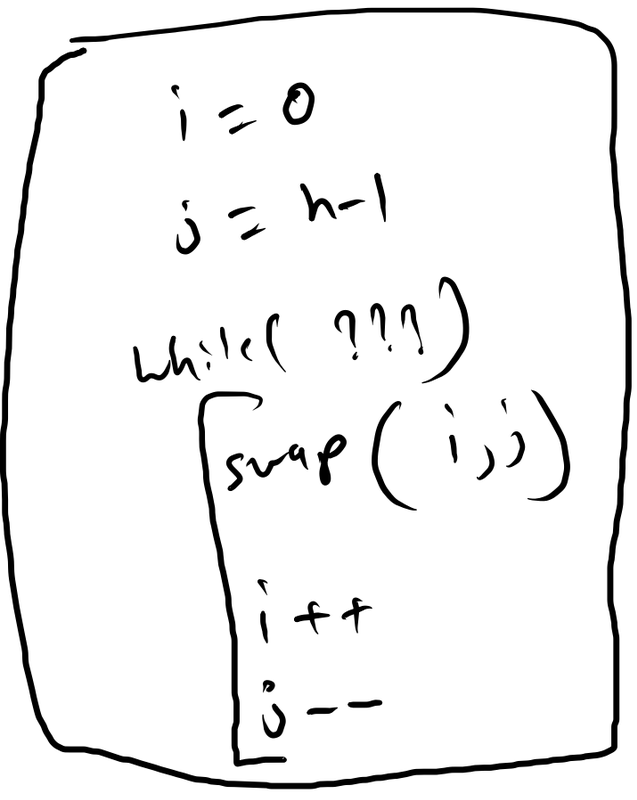



```
i = 0
j = h-1

while( ???)
    [ swap( i,j)

      i++
      j--
```



answer
is
original

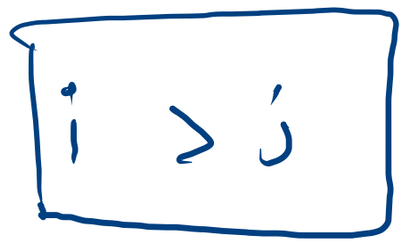




[e d c b a]

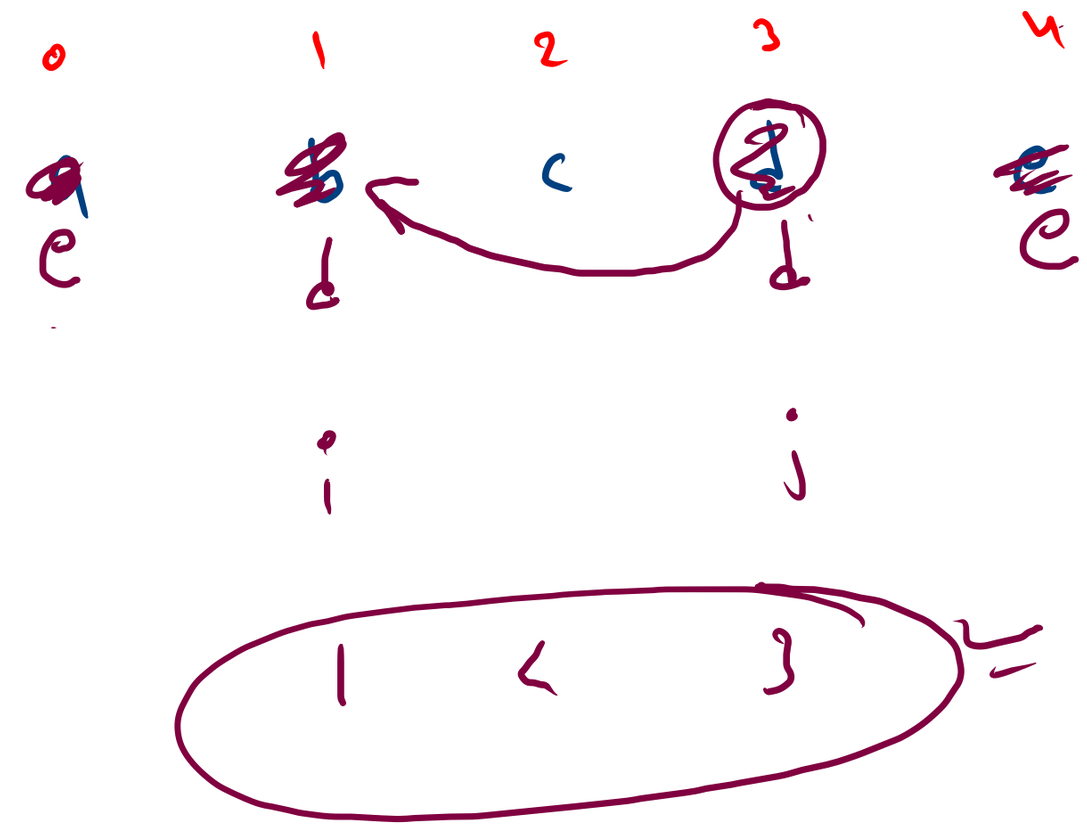
[i < j

[i == j



wrong

```java
public static void reverse(int[] a){

    int i=0;
    int j=a.length-1;

    while(i<j){
        a[i] = a[j];
        a[j] = a[i];

        i++;
        j--;
    }
```



```java
public static void reverse(int[] a){

    int i=0;
    int j=a.length-1;

    while(i<j){
        int tmp = a[i];
        a[i] = a[j];
        a[j] = tmp;

        i++;
        j--;
    }
}
```

swap

0 1 2 3 4

a b c d e

e d i b a

i j

tmp b

0	1	2	3	4	5
i		j			

(a.length-1)/2

5 -1

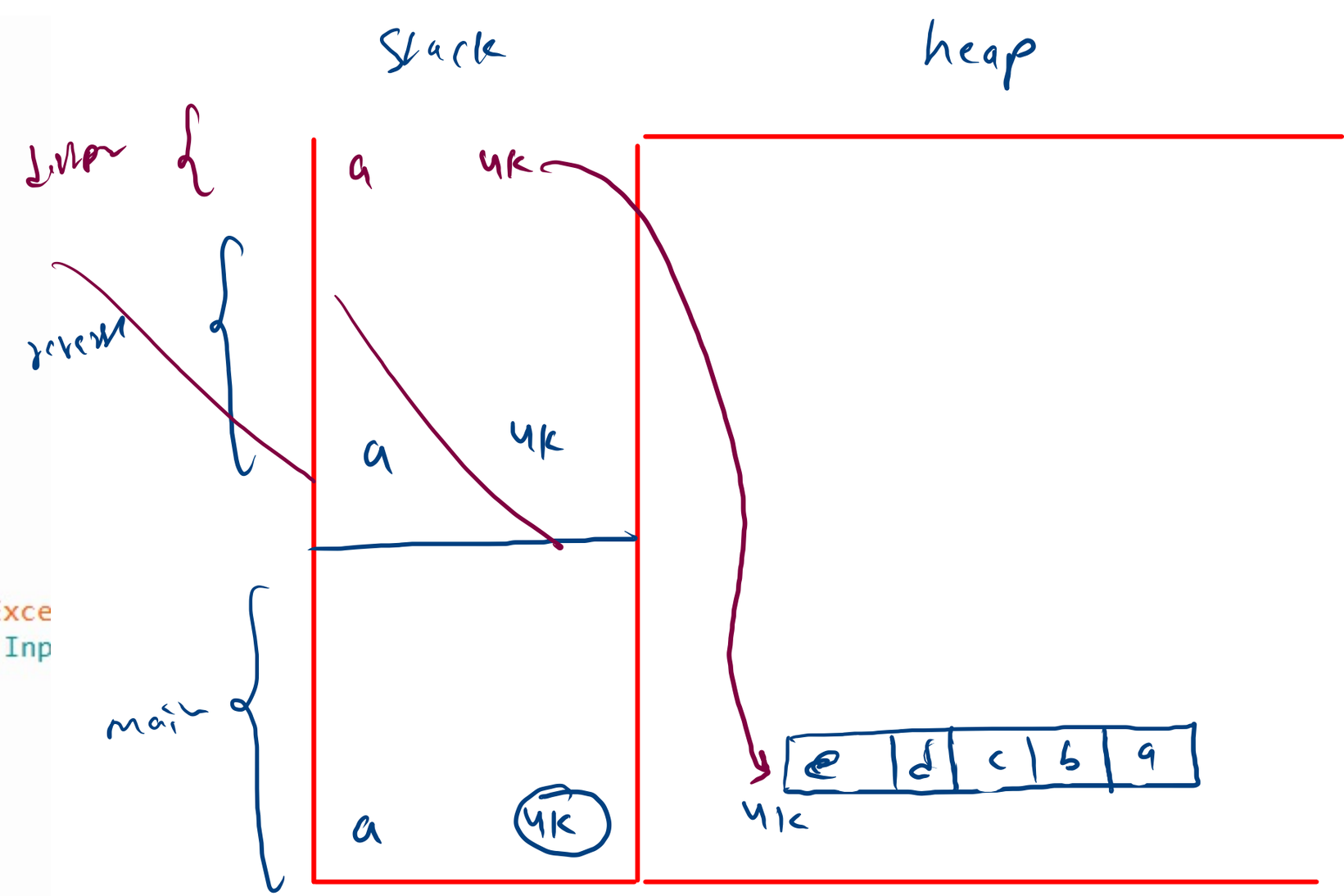
4 /2 = 2

(6-1)/2

5/2

2

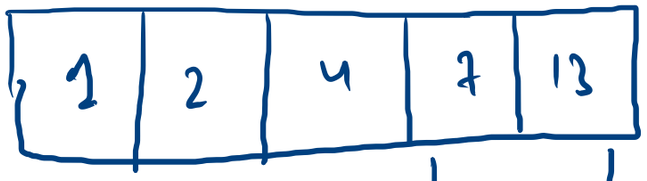
```java
public static void reverse(int[] a){

    int i=0;
    int j=a.length-1;

    while(i<j){
        int tmp = a[i];
        a[i] = a[j];
        a[j] = tmp;

        i++;
        j--;
    }
}

ublic static void main(String[] args) throws Exce
   BufferedReader br = new BufferedReader(new Inp

  int n = Integer.parseInt(br.readLine());
  int[] a = new int[n];
  for(int i = 0; i < n; i++){
     a[i] = Integer.parseInt(br.readLine());
  }

  reverse(a);
  display(a);
}
```

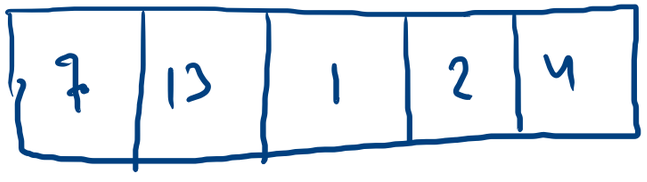


h = 5

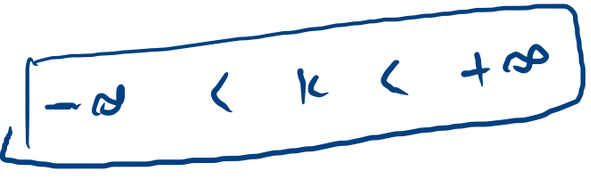
k = 2

arr →



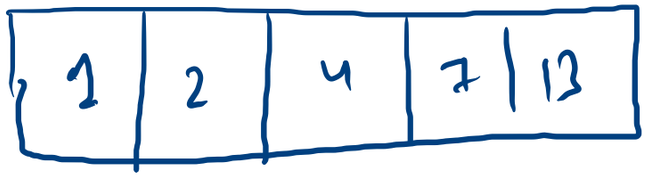


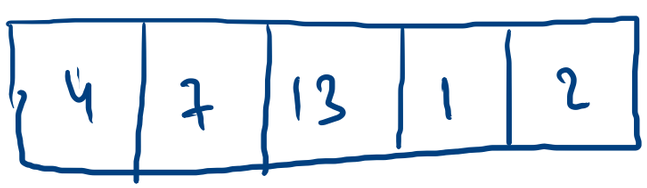
h x k

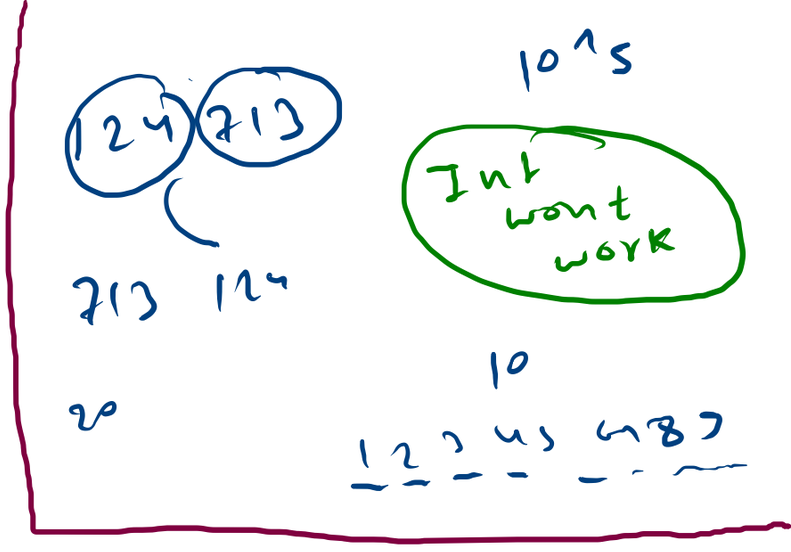


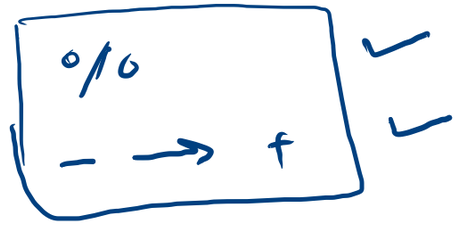
h = 5

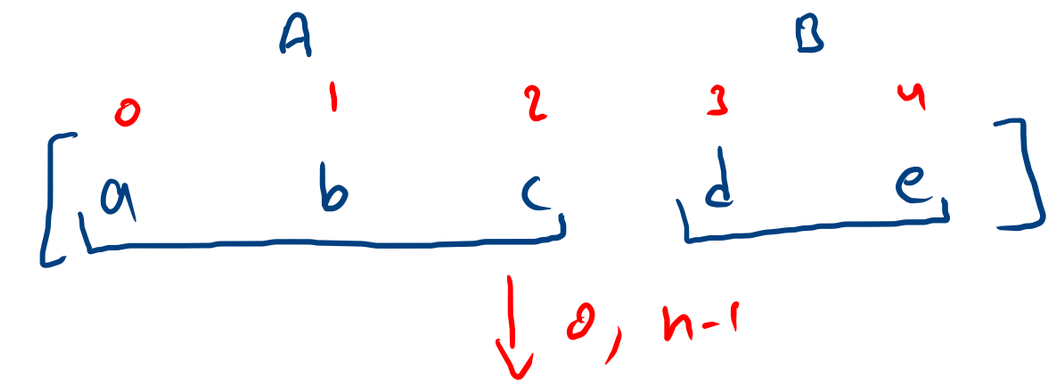
k = -2



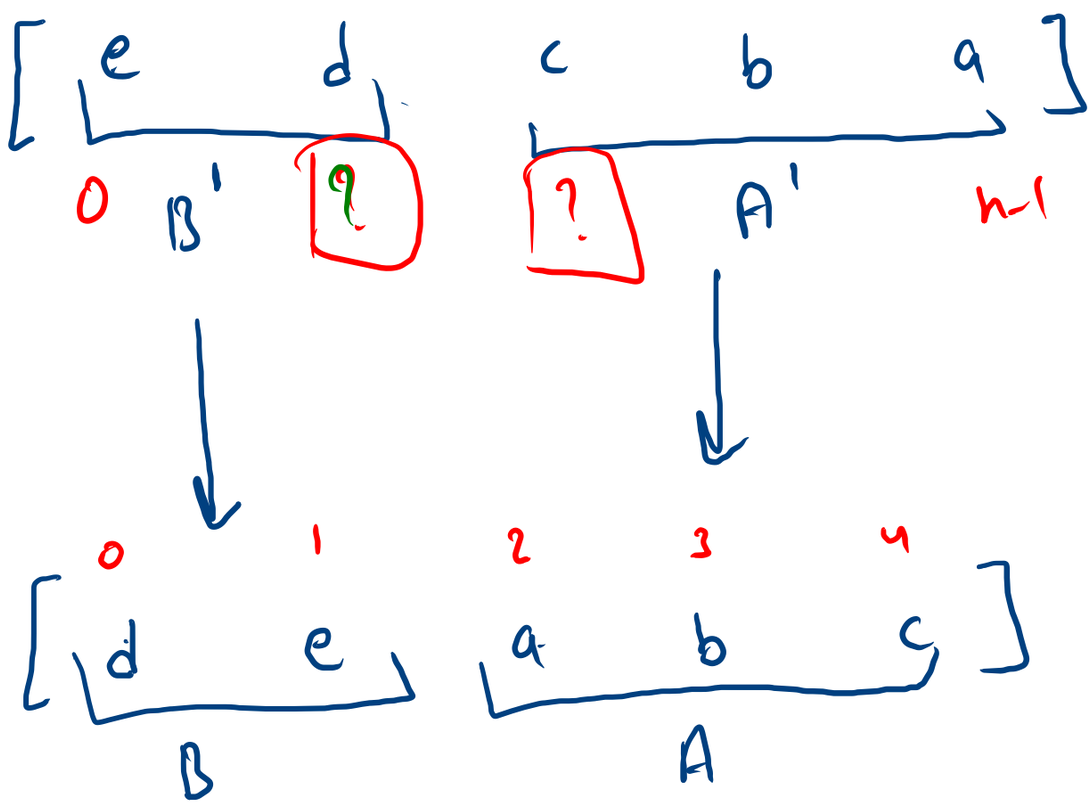








$k = 2$



0, n-1 full

0, k-1 B

k, n-1 A

$AB \rightarrow BA$

$(AB)' = B'A'$

$(B')(A')' = BA$

$A \rightarrow A' \rightarrow A$

i j

a b c d e → f e d c b a

i j

i i j

a b c d e f g

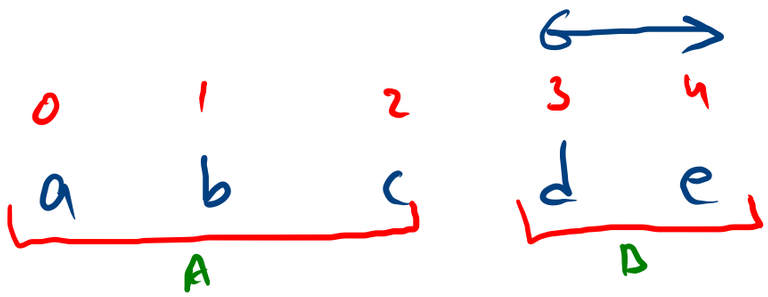
B A

e d c b a f g

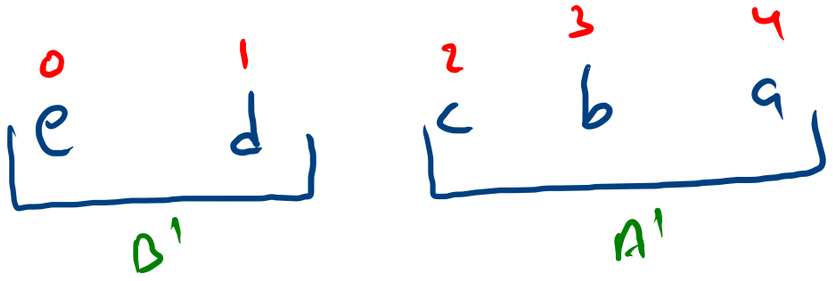
g f a b c d e

✡ a b c d e f g

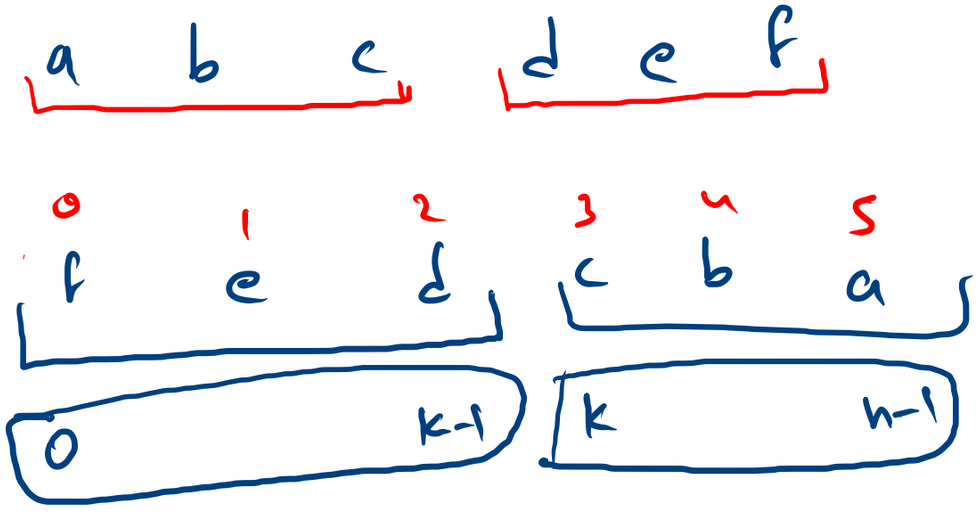
e d b a


0 1 2 3 4
a b c d e
A
D

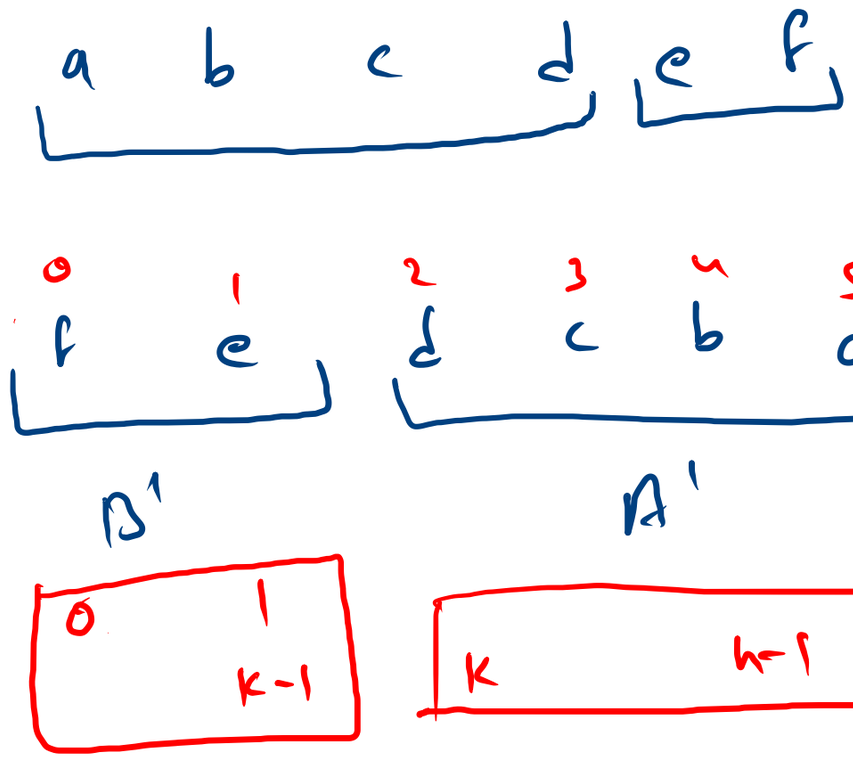
k = 2


0 1 2 3 4
e d c b a
D'
A'

k = 3


a b c d e f
0 1 2 3 4 5
f e d c b a
0 k-1
k h-1

k = 2


a b c d e f
0 1 2 3 4 5
f e d c b a
D'
A'
0 1
k-1
k h-1

H.W.

Inverse Of An Array

n = 5

arr →

0	1	2	3	4
3	0	2	1	4

i =	0	1	2	3	4
d =	3	0	2	1	4

0 .... n-1

0 4

inv →

0	1	2	3	4
1	3	2	0	4

i =	3	0	2	1	4
d =	0	1	2	3	4

n = 3

[10 (0) 20 (1) 30 (2)] ← print all sub-arrays

Continuous

0 — 10

0...1 — 10 20

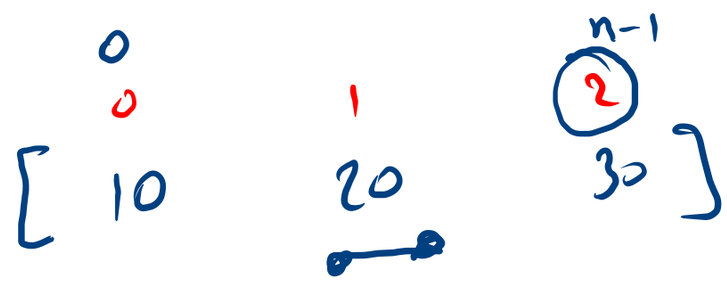
0...2 — 10 20 30 ✓

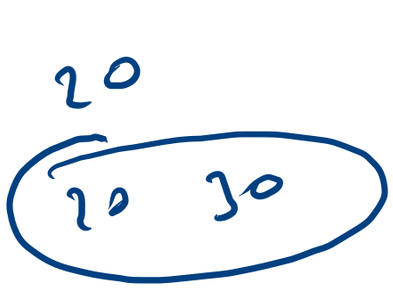
1 — 20

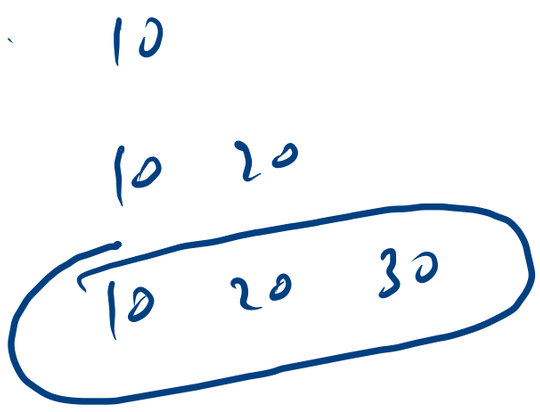
1...2 — 20 30

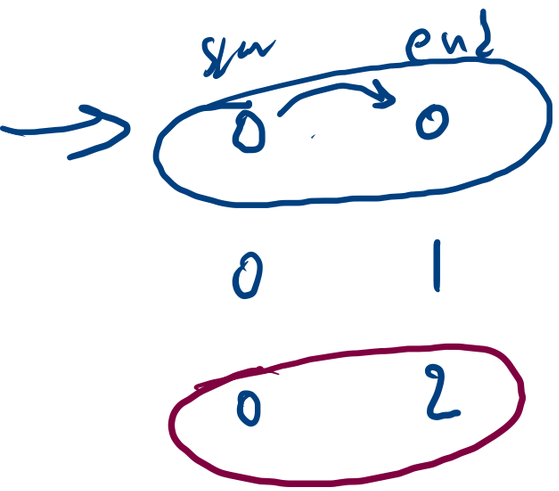
2...2 — 30

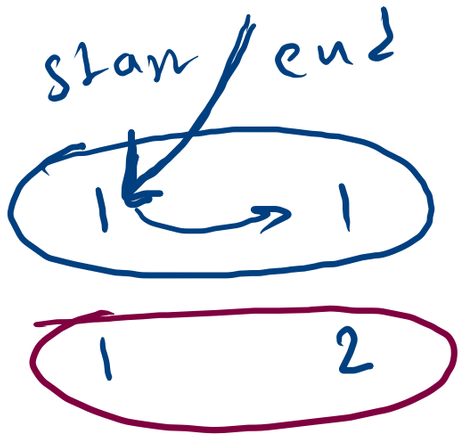
10 30 ✗

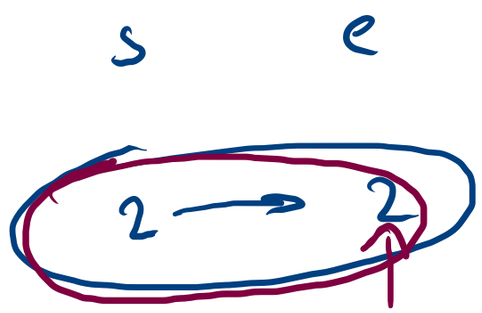












start 0 — do n-1 —

end start — do n-1 —

```
[ k ← start to end
      print ar[k]
println
```

```java
for(int i=0;i<n;i++){
    for(int j=i;j<n;j++){
        for(int k=i;k<=j;k++){
            System.out.print(arr[k]+"\t");
        }
        System.out.println();
    }
}
```

j<3

j<3

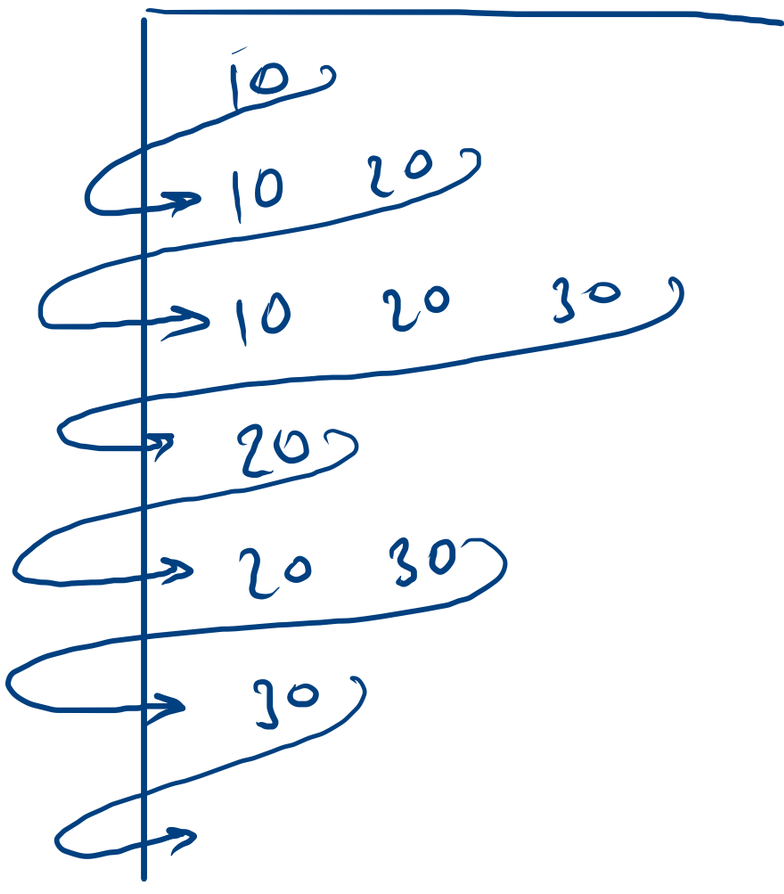
10 20 30

0 1 2

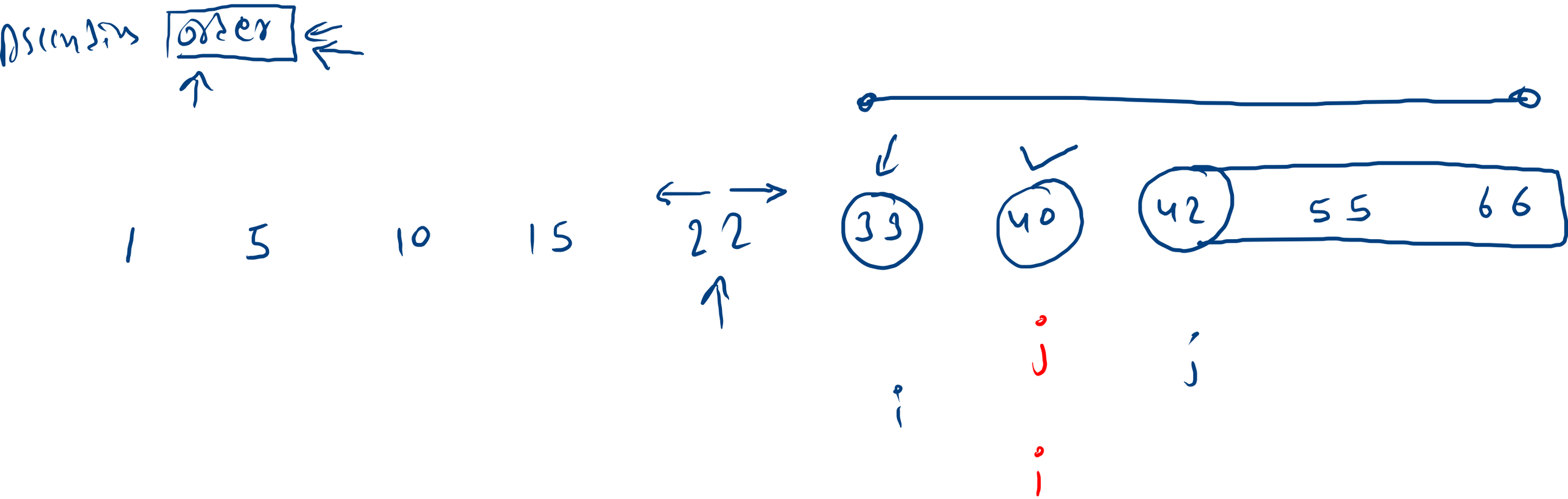
i = 0 1 2 3

j = 2 3

k = 2 3

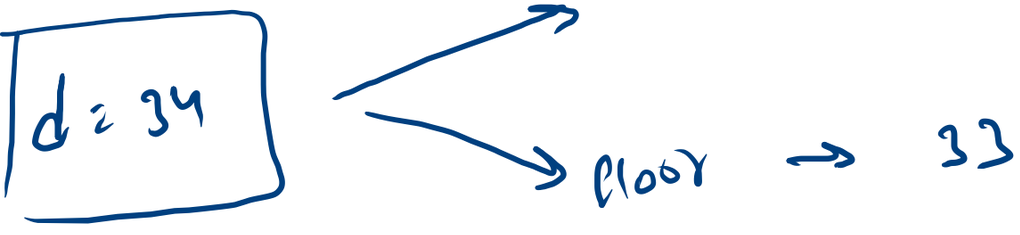


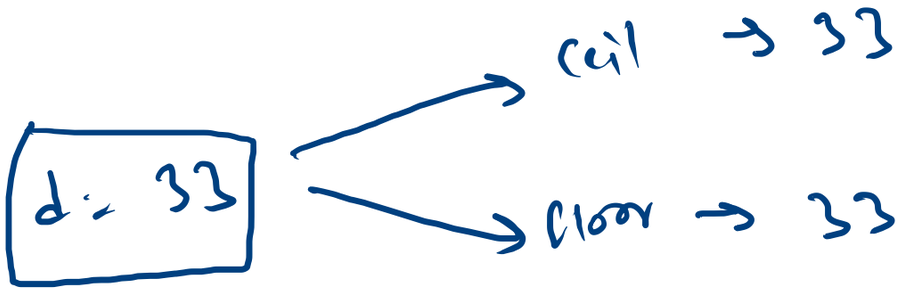
Descending order



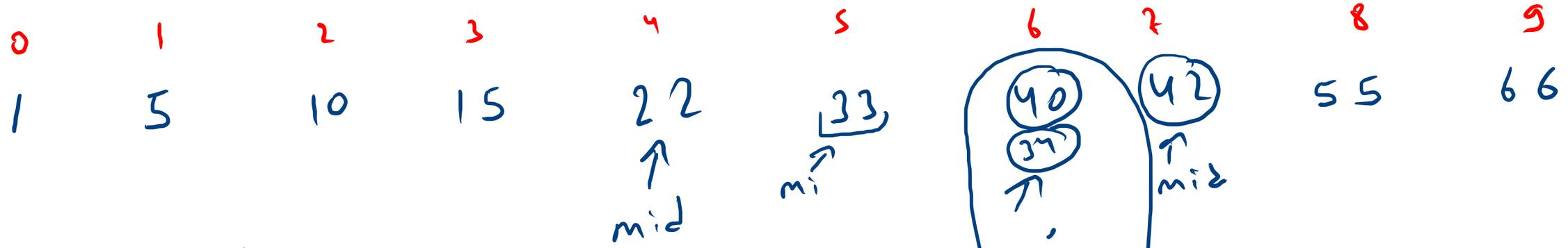
find 40

1 5 10 15 22 33 34 40 42 55 66





floor
ceil - 1



i ≤ j ✓

i > j ✗

d >

d < elm

d ==

```
mid = (i+j)/2
if(ar[mid] < d){
    floor = ar[mid]
    i = mid+1
} if(d < ar[mid]){
    ceil = ar[mid]
    j = mid-1
} else {
    flor = ar[mid]
    ceil = ar[mid]
} break;
```

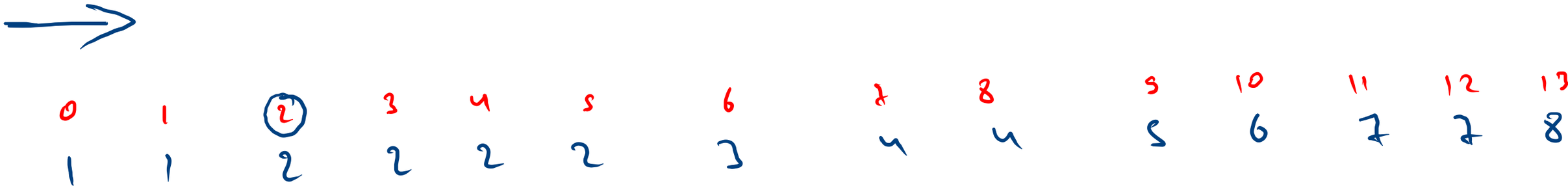
ceil biggr

flor smallr

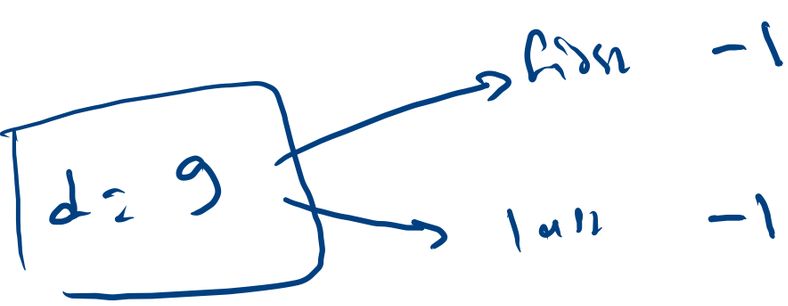
d = 34

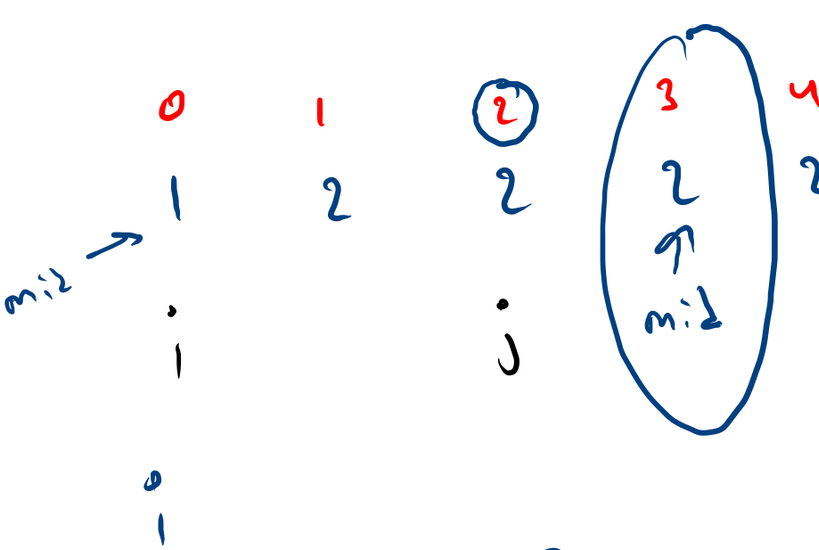
ceil ~~0~~ ~~42~~ 40

floor ~~0~~ ~~22~~ 33


0 1 2 3 4 5 6 7 8 9 10 11 12 13
1 1 2 2 2 2 3 4 4 5 6 7 7 8

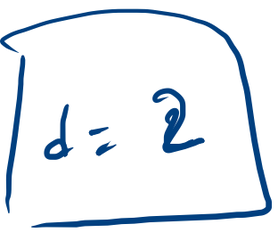

d = 2
first 2
last 5
mid
last


d = 9
first -1
last -1

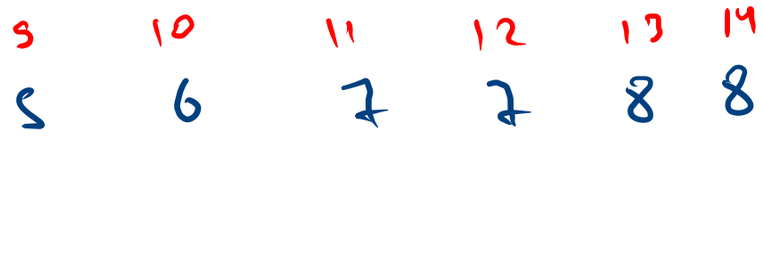


$\frac{0+1}{2} = 0$

2 < 4



first ≠ 3




```
mid = (i+j)/2
if( ar[mid] < d)
    i = mid+1
if( d < ar[mid])
    j = mid-1
else
    first = mid
    j = mid-1
```

d == [mid]