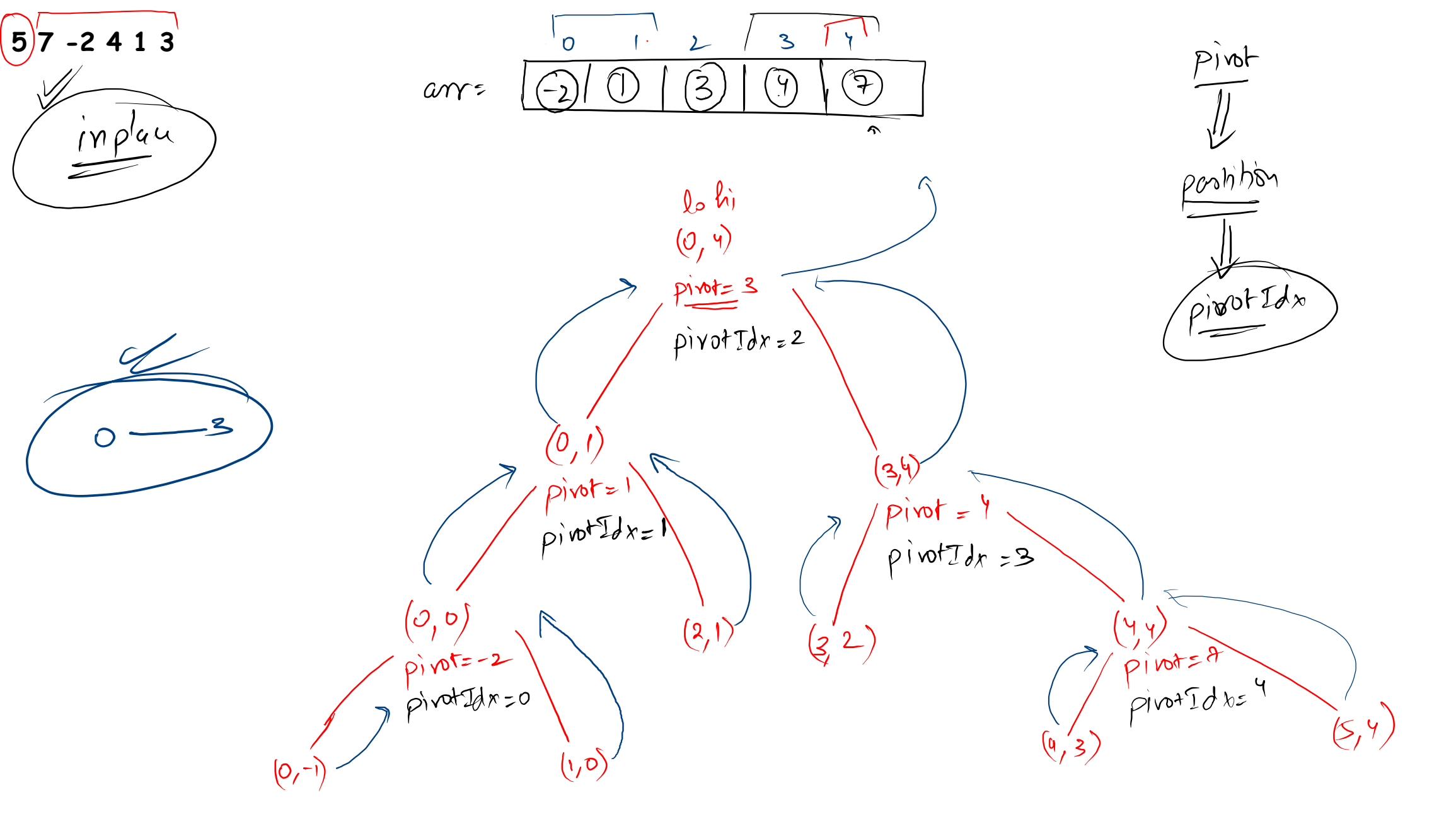
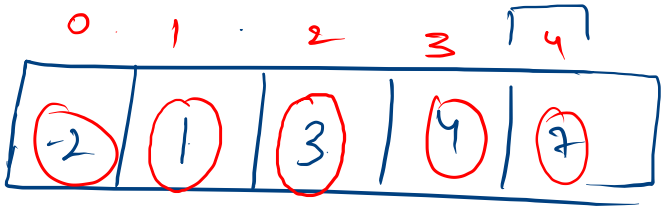
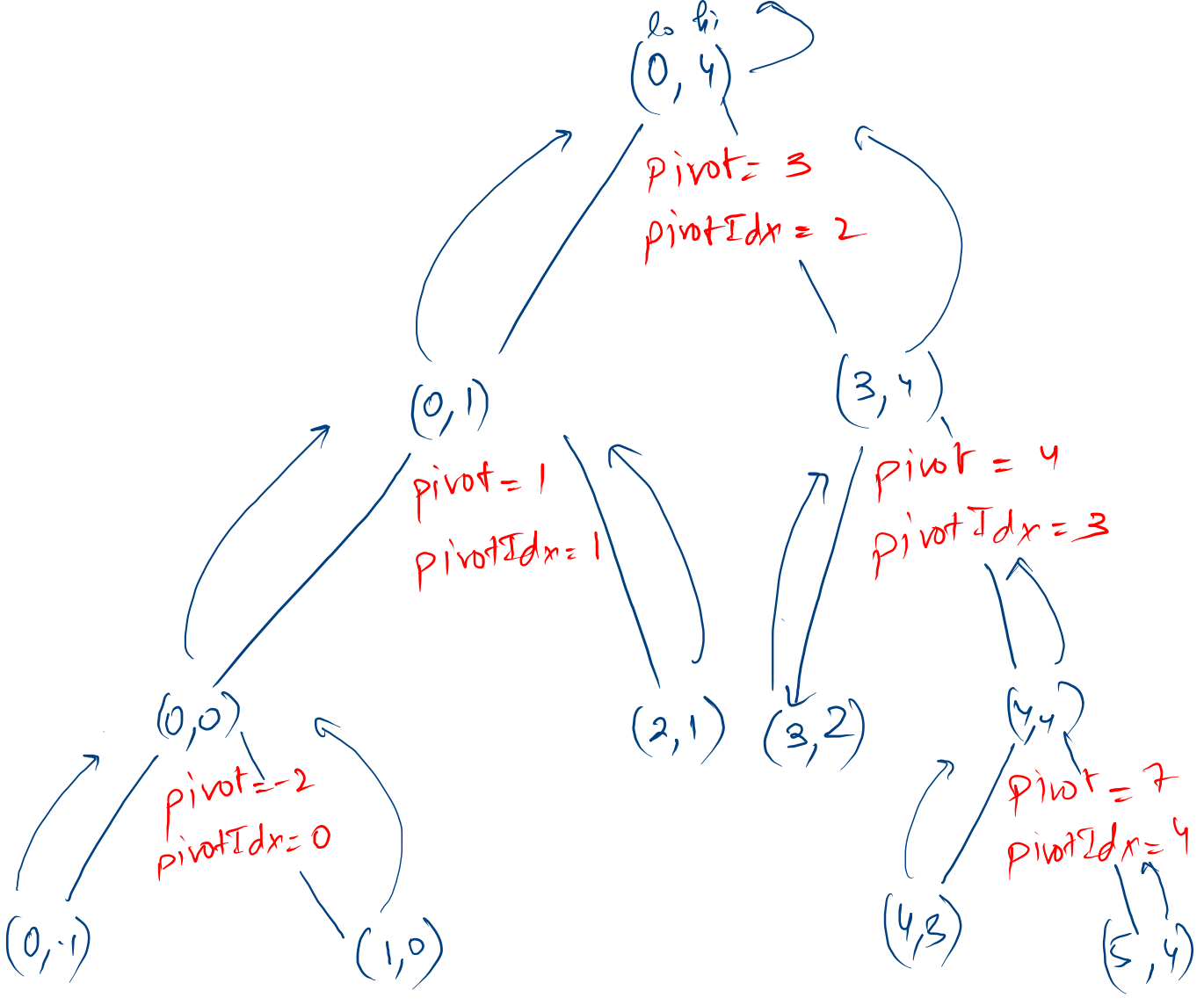

5 7 -2 4 1 3
inplace
arr =
0 1 2 3 4
-2 1 3 4 7
pivot
position
pivotIdx
0 3
lo hi
(0, 4)
pivot = 3
pivotIdx = 2
(0, 1)
pivot = 1
pivotIdx = 1
(3, 4)
pivot = 4
pivotIdx = 3
(0, 0)
pivot = -2
pivotIdx = 0
(2, 1)
(3, 2)
(4, 4)
pivot = 7
pivotIdx = 4
(0, -1)
(1, 0)
(4, 3)
(5, 4)


0
1
2
3
4
-2
1
3
4
7


lo hi
(0, 4)
pivot = 3
pivotIdx = 2
(0,1)
pivot = 1
pivotIdx = 1
(3, 4)
pivot = 4
pivotIdx = 3
(0,0)
pivot = -2
pivotIdx = 0
(2,1)
(3,2)
(4,4)
pivot = 7
pivotIdx = 4
(0,-1)
(1,0)
(4,3)
(5,4)

(0, 8)

pivot = 17

pivotIdx = 4

↓

(5, 8)

pivot = 18

pivotIdx = 5

↓

(6, 8)

pivot = 19

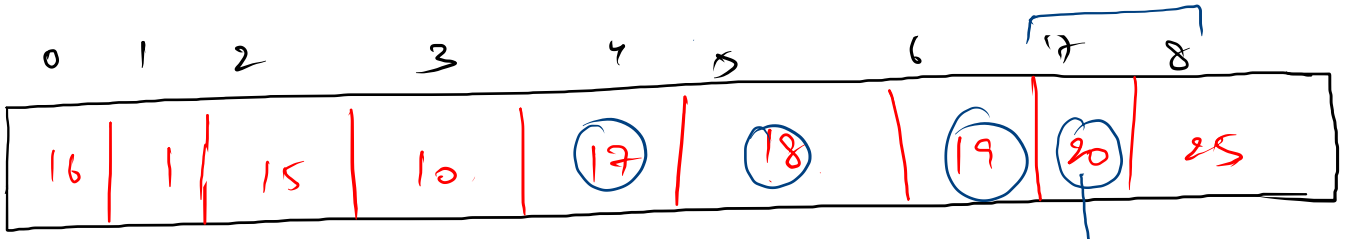
pivotIdx = 6

↓

(7, 8)

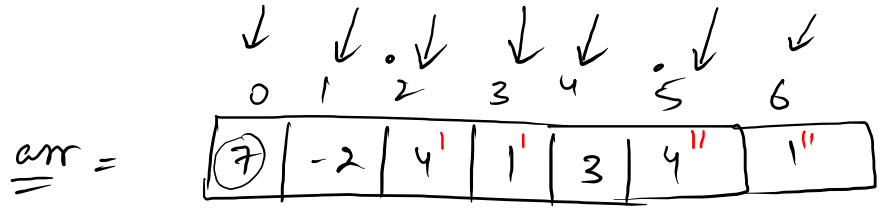
pivot = 20

pivotIdx = 7



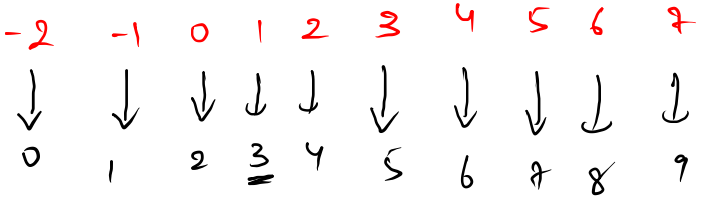
20 ✓

idx = 7 ✓

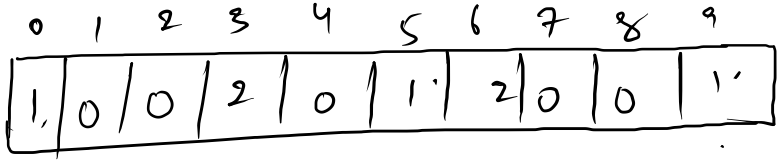


max = 7
min = -2

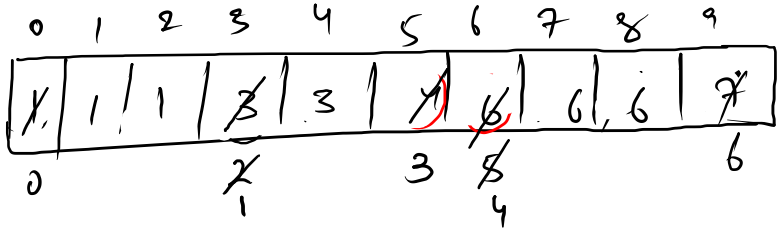
range ⇒ max - min + 1



① freq =

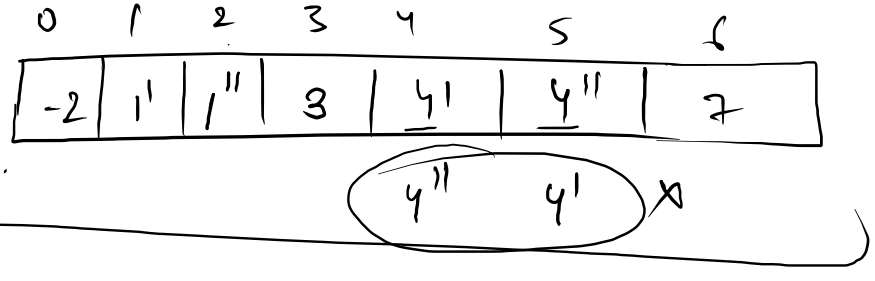


② psa =



① val
② pos
③ plac
④ idx

③



(1 2 4')
(4 4'')

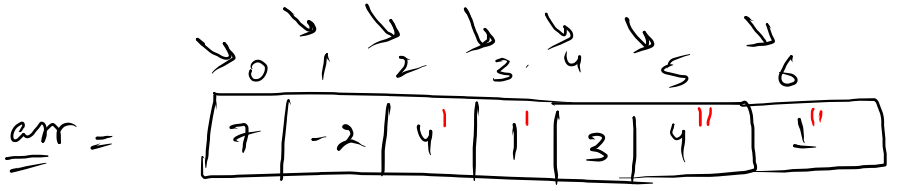
1 2 4'
4 4''

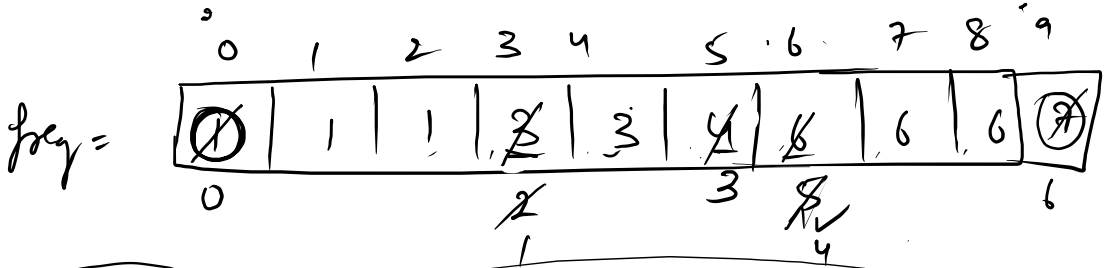
4 4''
1 2 4'

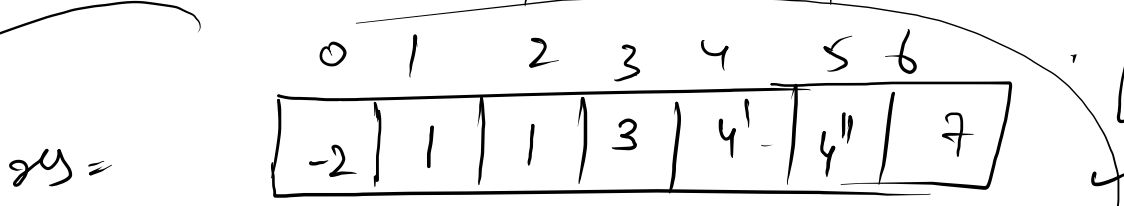
min = -2

max = 7

range ⇒ 10







-2 - (-2)

```java
int range = max-min+1;

int freq[] = new int[range];

for(int vl : arr){
    int pos = vl - min;
    freq[pos]++;
}

for(int i = 1 ; i < range ; i++){
    freq[i] += freq[i-1];
}

int res[] = new int[arr.length];

for(int i = arr.length-1; i >= 0; i--){
    int vl = arr[i]; // 7
    int pos = vl - min; // 9
    int place = freq[pos]; // 7
    res[place-1] = vl;
    freq[pos]--;
}
```

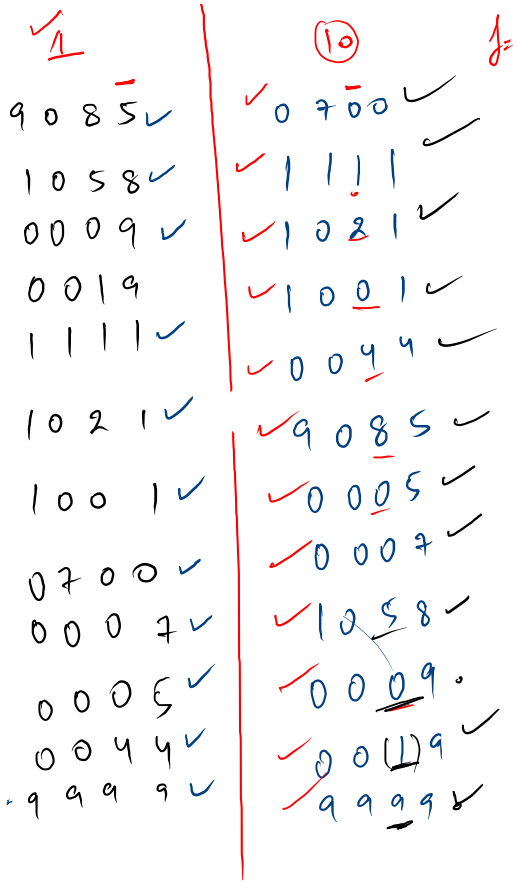
Radix

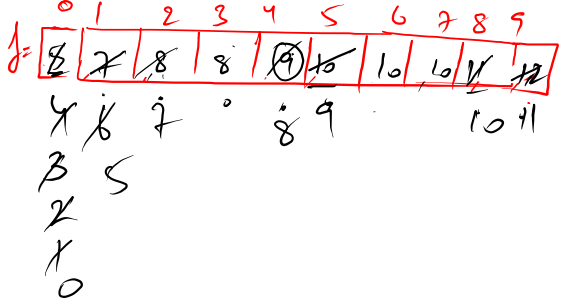
(Least Sig → Most Sig)

(9085 / 10) % 10 => 8

(d <= (num / exp) % 10)

10(2)/100 → 10 % 10 → 10



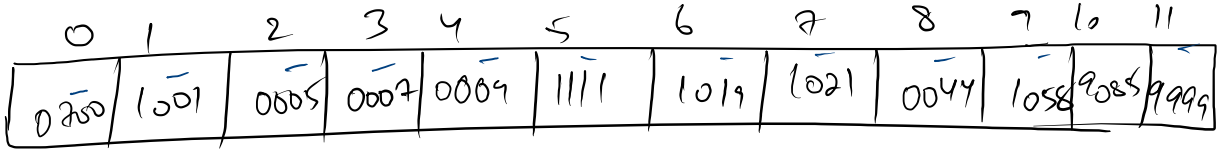


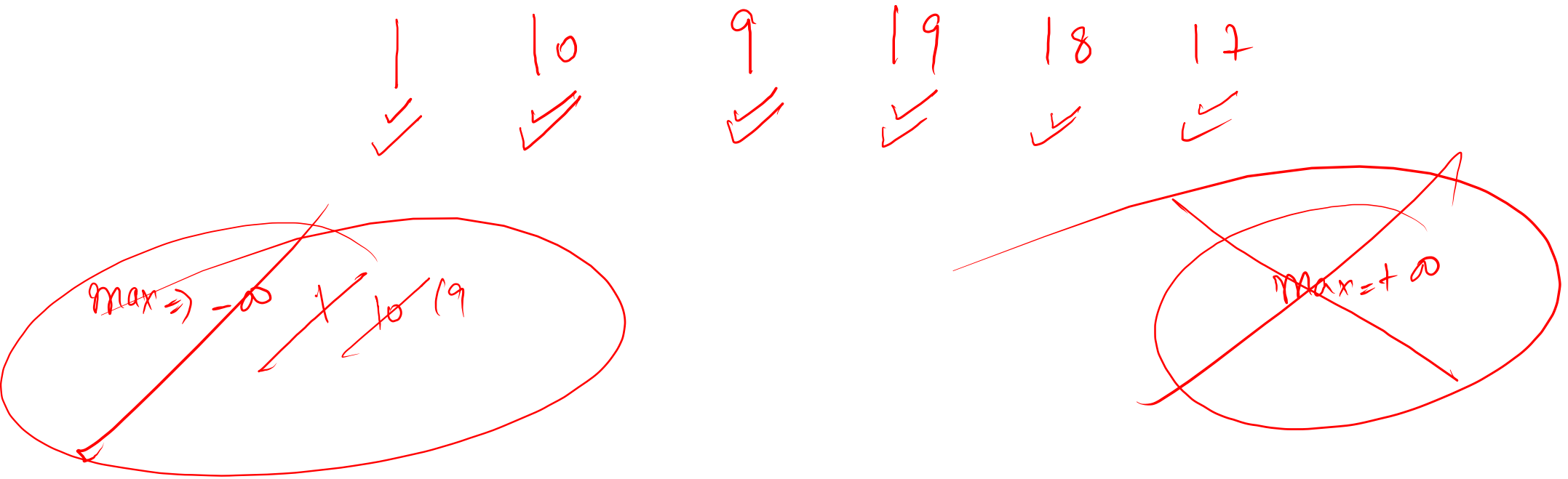

```
public static void countSort(int[] arr, int exp) {
  int freq[] = new int[10];
  for(int vl : arr){
      freq[(vl/exp)%10]++;
  }

  for(int i = 1 ; i < 10 ; i++){
      freq[i] += freq[i-1];
  }

  int res[] = new int[arr.length];
  for(int i = arr.length-1 ; i>= 0 ; i--){
      int vl = arr[i];
      int pos = (vl/exp)%10;
      int place = freq[pos];
      res[place-1] = vl;
      freq[pos]--;
  }

  for(int i = 0 ; i < arr.length ; i++){
      arr[i] = res[i];
  }
  System.out.print("After sorting on " + exp + " place -> ");
  print(arr);
}
```




1
10
9
19
18
17
max => -∞
1
10
19
max = +∞

DD MM YYYY

5
12041996
20101996
05061997
12041989
11081987

Count Dates

Ans → 3

H.W.

Sum

I

Day [32]

05 ~~06 1997~~
11 08 1987
12 04 1996
12 04 1989
20 10 1996

II

Month [13]

12 04 ~~1996~~
12 04 1989
05 06 1997
11 08 1987
20 10 1996

III

Year [2501]

11 08 (1987)
12 04 1989
12 04 1996
20 10 1996
05 06 1997

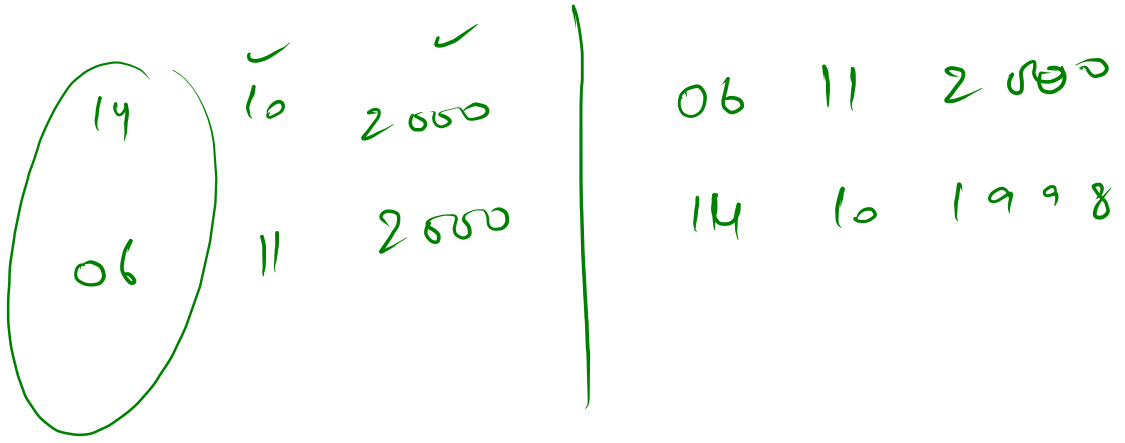
day → (date / 1000000)

month = (date / 10000) % 100

year ← date % 10000

(12 / 10) % 10 → 2

[ ]


14 10 2000
06 11 2000
06 11 2000
14 10 1998