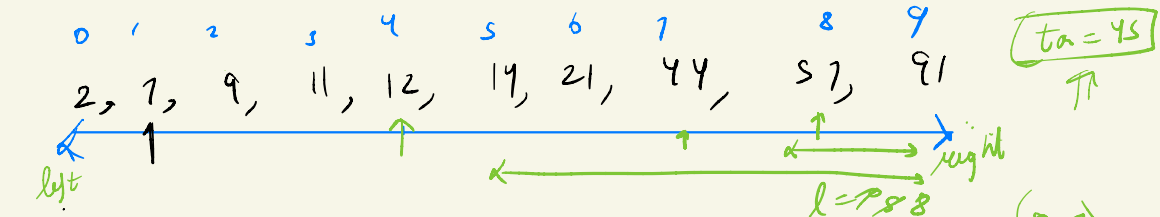


tar = 45

mid = 5+9/2 = 7

mid = 9+8/2 → 8

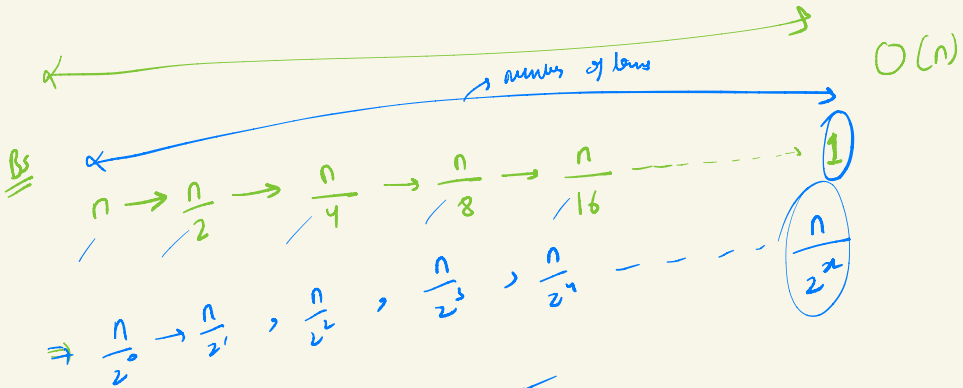
```java
public static int binarySearch(int[] arr, int tar){
    int n=arr.length;

    int left=0;
    int right=n-1;

    while(){
        int mid=(left+right)/2;

        if(arr[mid]==tar){
            return mid;
        } else if(arr[mid]<tar){
            left=mid+1;
        } else {
            right=mid-1;
        }
    }

    return -1;
}
```



number of terms

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

$\log n = \log 2^x$ $\quad \log_2 2 = 1$

$\log_2 n = x \log_2 2$

$x = \log_2 n$

0 1 2 3 4 5 6 7 8 9

91, 51, 44, 21, 19, 12, 11, 9, 7, 2

(44)

tar = 9

if (tar < arr(mid)) {
    left = mid+1;
}
if (tar > arr(mid)) {
    right = mid-1
}

Ans

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

{1, 1, 3, 3, 3, 3, 4, 4, 4, 5, 5, 8, 8, 8, 8, 8, 8, 8, 10}

tar = 8

left = ~~0 10 15 17~~ 18

right = 18 17

mid = ~~9 14 16~~ 17

lo = ~~14 16~~ 17

if (arr(mid) == target) {
    lo = mid
    left = mid+1
}
if (arr(mid) < target) {
    left = mid+1;
}

0 → 0 0 0 0 → 2^0

1 → 0 0 0 1

2 → 0 0 1 0 → 2^1

3 → 0 0 1 1

4 → 0 1 0 0 → 2^2

5 → 0 1 0 1

6 → 0 1 1 0

7 → 0 1 1 1

8 → 1 0 0 0 → 2^3

9 → 1 0 0 1

10 → 1 0 1 0

11 → 1 0 1 1

12 → 1 1 0 0

13 → 1 1 0 1

14 → 1 1 1 0

15 → 1 1 1 1

16 → 1 0 0 0 0 → 2^4

Convert a decimal number to binary

Ques

17 ./. 2

⇓

n = 17.

n = n/2

2	17	1
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

String binary = 1

binary = '0' + binary

⇒ 01

⇒ 001

0001

10001

n = 23 → 5 → 2 → 1 → 0 | binary = "" → "1" → "11" → "111" → "1011"

rem = 1 % 2 = 01

"0" + "111"

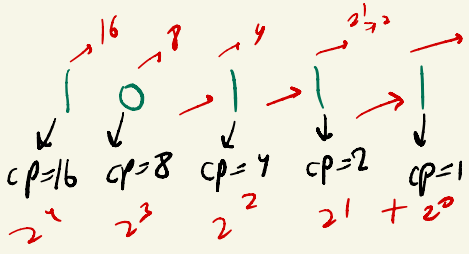
2 | 23	
2 | 11	1
5	1

```java
public static String decimalToBinary(int n){
    String binary="";

    while(n>0){
        int rem=n%2;

        binary= rem + binary;
        n=n/2;
    }
    return binary;
}
```

Ques Given a binary string, change it to decimal number.



$2^0 + 2^1 + 2^2 + 2^4$

$\downarrow$

$1 + 2 + 4 + 16 \Rightarrow 23$

cp = 1 2 4 8 16 32

ans = 0 + 2 + 4 + 16 = 23

```java
public static int binaryToDecimal(String binary){
    int n=binary.length();

    int curr_pow=1;
    int ans=0;

    for(int i=n-1; i>=0; i--){
        char ch=binary.charAt(i);

        if(ch=='1'){
            ans=ans+curr_pow;
        }

        curr_pow=curr_pow*2;
    }

    return ans;
}
```

power of 2