

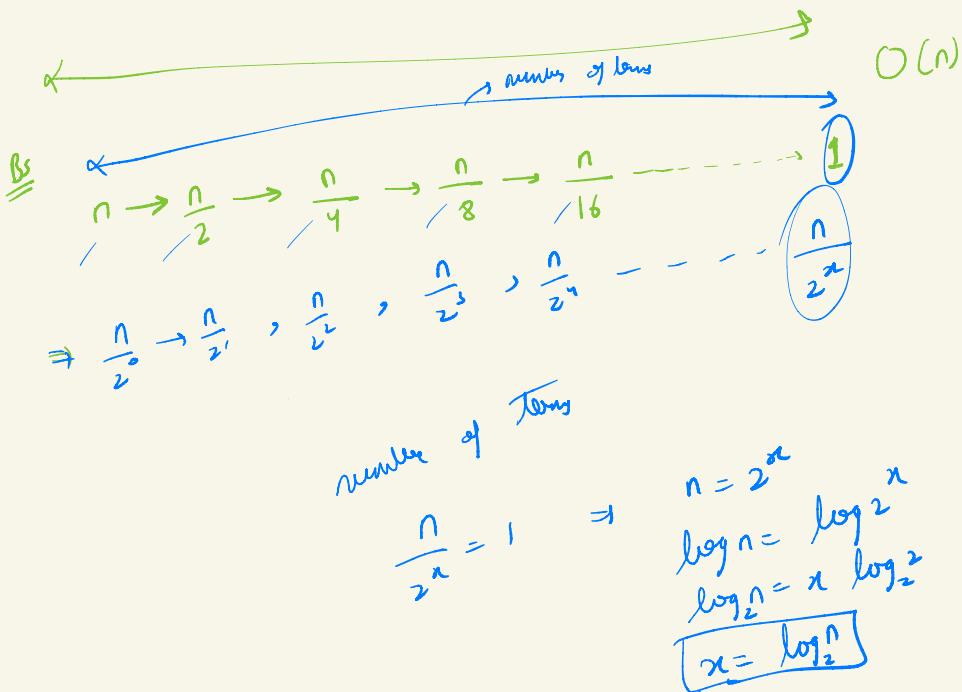
```
public static int binarySearch(int[] arr, int tar){
    int n=arr.length;

    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;
}
```



$\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 91, 51, 44, 21, 14, 12, 11, & 9, 7, 2 \end{matrix}$

(44)

$\text{tar} = 9$

$\left. \begin{array}{l} \text{if } (\text{tar} < \text{arr}(\text{mid})) \{ \\ \quad \text{left} = \text{mid} + 1; \end{array} \right\}$

$\left. \begin{array}{l} \text{if } (\text{tar} > \text{arr}(\text{mid})) \{ \\ \quad \text{right} = \text{mid} - 1 \end{array} \right\}$

Ans

$\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\ 1, 1, 3, 3, 3, 3, 4, 4, 4, 5, 5, 8, 8, 8, 8, 8, 8, 8, 10 \end{matrix}$

$\text{tar} = 8$

$\text{let } t = 8$
 $\text{left} = 1$
 $\text{right} = 18$
 $\text{mid} = 9$
 $10 / 11$

$\text{tar} = 14 / 15 / 17$

$\left. \begin{array}{l} \text{if } (\text{arr}(\text{mid}) == \text{target}) \{ \\ \quad \text{left} = \text{mid}; \end{array} \right\}$

$\text{left} = \text{mid}$

$\text{left} = \text{mid} + 1$

$\left. \begin{array}{l} \text{if } (\text{arr}(\text{mid}) < \text{target}) \{ \\ \quad \text{left} = \text{mid} + 1; \end{array} \right\}$

left

$$0 \rightarrow \underline{0} \quad \underline{0} \quad \underline{0} \quad \underline{0} \rightarrow 2^0$$

$$1 \rightarrow \underline{0} \quad \underline{0} \quad \underline{0} \quad -$$

$$2 \rightarrow \underline{0} \quad \underline{0} \quad \underline{1} \quad \underline{0} \rightarrow 2^1$$

$$3 \rightarrow \underline{0} \quad \underline{0} \quad \underline{1} \quad \underline{1}$$

$$4 \rightarrow \underline{0} \quad \underline{1} \quad \underline{0} \quad \underline{0} \rightarrow 2^2$$

$$5 \rightarrow \underline{0} \quad \underline{1} \quad \underline{0} \quad \underline{1}$$

$$6 \rightarrow \underline{0} \quad \underline{1} \quad \underline{1} \quad \underline{0}$$

$$7 \rightarrow \underline{0} \quad \underline{1} \quad \underline{1} \quad \underline{1}$$

$$8 \rightarrow \underline{1} \quad \underline{0} \quad \underline{0} \quad \underline{0} \rightarrow 2^3$$

$$9 \rightarrow \underline{1} \quad \underline{0} \quad \underline{0} \quad \underline{1}$$

$$10 \rightarrow \underline{1} \quad \underline{0} \quad \underline{1} \quad \underline{0}$$

$$11 \rightarrow \underline{1} \quad \underline{1} \quad \underline{0} \quad \underline{1}$$

$$12 \rightarrow \underline{1} \quad \underline{1} \quad \underline{0} \quad \underline{0}$$

$$13 \rightarrow \underline{1} \quad \underline{1} \quad \underline{0} \quad \underline{1}$$

$$14 \rightarrow \underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{0}$$

$$15 \rightarrow \underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{1}$$

$$16 \rightarrow \underline{1} \quad \underline{0} \quad \underline{0} \quad \underline{0} \rightarrow 2^4$$

Convert a decimal number to binary

Ques

$$17 \div 2$$

↓

$$n=17.$$

$$n=n/2$$

$$\begin{array}{r} 2 | 17 \rightarrow 0 \\ \hline 2 | 8 \checkmark \quad 1 \\ \hline 2 | 4, \quad 0 \\ \hline 2 | 2, \quad 0 \\ \hline 2 | 1, \quad 0 \\ \hline \quad 0 \rightarrow 1 \end{array}$$

$$\text{String binary} = 1$$

$$\begin{aligned} \text{binary} &= '0' + \text{binary} \\ &\Rightarrow 01 \end{aligned}$$

$$\Rightarrow \underline{0} \quad 0 \quad 1$$

$$\underline{\underline{0}} \quad 0 \quad 0 \quad 1$$

$$\underline{\underline{\underline{1}}} \quad 0 \quad 0 \quad 0 \quad 1$$

$$n = \cancel{2^3} + \cancel{8} \times 10 / \text{binary} = "1" \cancel{+} \cancel{1} \cancel{+} \underline{\underline{1011}}$$

$$\text{num} = 1 \cdot 2 = 01$$

$$\begin{array}{r} 2 | 23 \\ 2 | 11 \\ \hline 5, 1 \end{array}$$

```
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.

$$\begin{array}{ccccccc}
 & 16 & 8 & 4 & 2 & 1 & 2^0 = 1 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 cp=16 & cp=8 & cp=4 & cp=2 & cp=1 & & \\
 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & \\
 & 1 & 1 & 0 & 1 & & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & & \\
 & 2^0 + 2^1 + 2^2 + 2^4 & & & & & \\
 & 1 + 2 + 4 + 16 = 23 & & & & &
 \end{array}$$

$$\begin{array}{l}
 cp=\cancel{16} + \cancel{8} + \cancel{4} + \cancel{2} \\
 \text{Any} = \cancel{16} + \cancel{8} + \cancel{4} + 2^3
 \end{array}$$

```
public static int binaryToDecimal(String binary){
    int n=binary.length();
    int curr_pow=1; → power of 2
    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;
}
```

Ques Given one sorted array, find total number of element = tar.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1, 2, 3, 3, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 8, 8, 8, 10																			

tar
output = ?

$$f_i = 14$$

$$l_i = 18 \quad B^{-14+1}$$

0	1	2	3	4	5	6
1, 2, 4, 5, 6, 8, 10	↑	↑	↑	↑	↑	↑

$$l=9, 4, 6 \text{ (7)}$$

$$r=6$$

$$\text{tar} = 11$$

```
public static int find_index_of_just_greater(int[] arr, int tar) {
    int left=0;
    int right=arr.length-1;

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

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

    return left;
}
```

$\rightarrow s1 = bbaafcd$ ✓

$\rightarrow s2 = ghiz2*$

a → x²

b → x²

f → 1

c → 1

d → 1

```
public static String twoStrings(String s1, String s2) {
    int[] fre=new int[26];

    for(int i=0; i<s1.length(); i++){
        char ch=s1.charAt(i);

        int idx=ch-'a';
        fre[idx]++;
    }

    for(int i=0; i<s2.length(); i++){
        char ch=s2.charAt(i);

        int idx=ch-'a';
        if(fre[idx]>0){
            return "YES";
        }
    }

    return "NO";
}
```

$$\text{area} = \frac{1}{2} \times b \times h$$

$$h=17 \\ \text{area}=100$$

$$h = \frac{2 \times \text{area}}{b}$$

$$\Rightarrow \frac{2 \times 100}{17} = \frac{200}{17} = 11.76$$

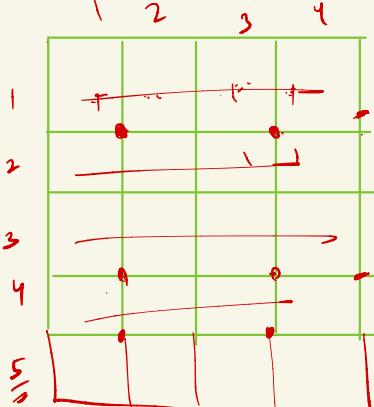
floor
11 $\leftarrow 11.76 \rightarrow 12$

$$m = \underline{\underline{s}}$$

$$\left(\frac{m+1}{2} \right)$$

$$\left(\frac{n+1}{2} \right)$$

$$\left(\frac{n}{2} \right)$$



$$\left(\frac{n}{2} \right)$$

$$\left(\frac{20}{2} \right)$$

$$\left(\frac{n+m}{2} \right)$$

$$\frac{n}{2} \times \left(\frac{m+1}{2} \right)$$

$$2 \times 3 = 6$$

$$\left[\left(\frac{n+1}{2} \right) \times \left(\frac{m+1}{2} \right) \right] =$$

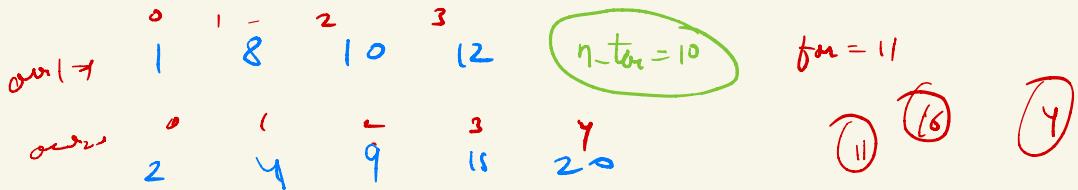
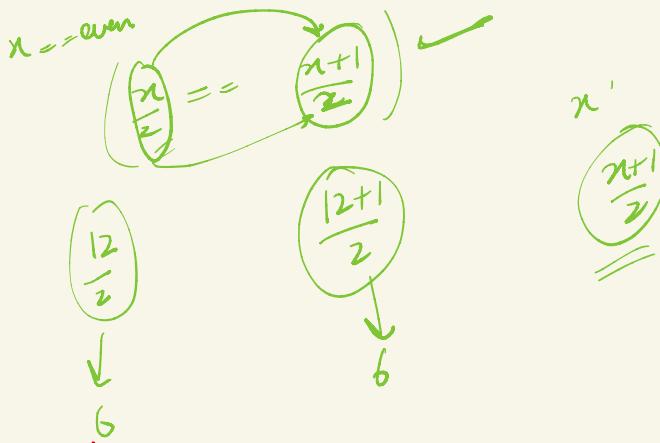
$$m = \underline{\underline{s}}$$

$$\left(\frac{s}{2} \right) \Rightarrow 2$$

number of rows to be covered = $\frac{1}{2}$ even
 number of columns " " " " $\Rightarrow \frac{m}{2}$ even

$$\left(\frac{n}{2} \right) = \left(\frac{n+1}{2} \right)$$

$$\left(\frac{n+1}{2} \right)$$



$$diff = 2^3 - 1 \quad n_1 = 1 \quad n_2 = 9$$

$$i=0 \quad c - diff = 10 \\ j = 0 \times 3 \times 4 \quad curr = x_1 + x_2 \\ curr = 16 + 21$$

```

public static void getClosest(int[] arr1, int[] arr2, int tar){
    int diff=Integer.MAX_VALUE;
    int x1=-1;
    int x2=-1;

    for(int i=0; i<arr1.length; i++){
        for(int j=0; j<arr2.length; j++){
            int csum= arr1[i] + arr2[j];

            int curr_diff = Math.abs(csum - tar);

            if(curr_diff < diff){
                x1=arr1[i];
                x2=arr2[j];
                diff=curr_diff;
            }
        }
    }

    System.out.println(x1+" "+x2);
}

```

x_1	1	2	3	4	5	6
x_2	1	8	9	15	17	

$\text{tar} \Rightarrow 18$

$$-7 \quad 3 \quad (-5 \quad 2 \quad 4)$$

$\ell_{\max} \rightarrow$

-7	3	3	3	4
0	1	2	3	4

$\ell_{\min} \rightarrow$

-1	-1	-1	-1	-1
0	1	2	3	4

$\mu_{\max} \rightarrow$

4	4	4	4	4
---	---	---	---	---

$\mu_{\min} \rightarrow$

-1	-5	-5	2	4
----	----	----	---	---

ℓ_{\max}
 \hookrightarrow left position

ℓ_{\min}
 \hookrightarrow left negative

$O(n^2)$

\downarrow
 $O(n)$

$$\begin{aligned} &\Rightarrow \cancel{\ell_{\max}(i-1)} \times \cancel{\mu(i)} \times \cancel{\ell_{\min}(i+1)} \\ &\Rightarrow \cancel{\ell_{\min}(i-1)} \times \cancel{\mu(i)} \times \cancel{\ell_{\min}(i+1)} \\ &\Rightarrow \cancel{\mu(i)} \times \ell_{\max}(i-1) \times \cancel{\ell_{\min}(i+1)} \\ &\Rightarrow \cancel{\mu(i)} \times \ell_{\min}(i-1) \times \cancel{\ell_{\max}(i+1)} \end{aligned}$$

neg neg pos