Day-11 String

H.W.

① $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

②

$\begin{bmatrix} 1 & 5 & 7 \\ 9 & 10 & 13 \\ 15 & 18 & 21 \end{bmatrix}$

Search = 27

```
for(int i=0; i<row; i++)
{
    for(int j=0; j<clm; j++)
    {
        if(arr[i][j]==search)
        {
            S.o.p(-->
            fun
        } return
    }
}
return false
```

n×n

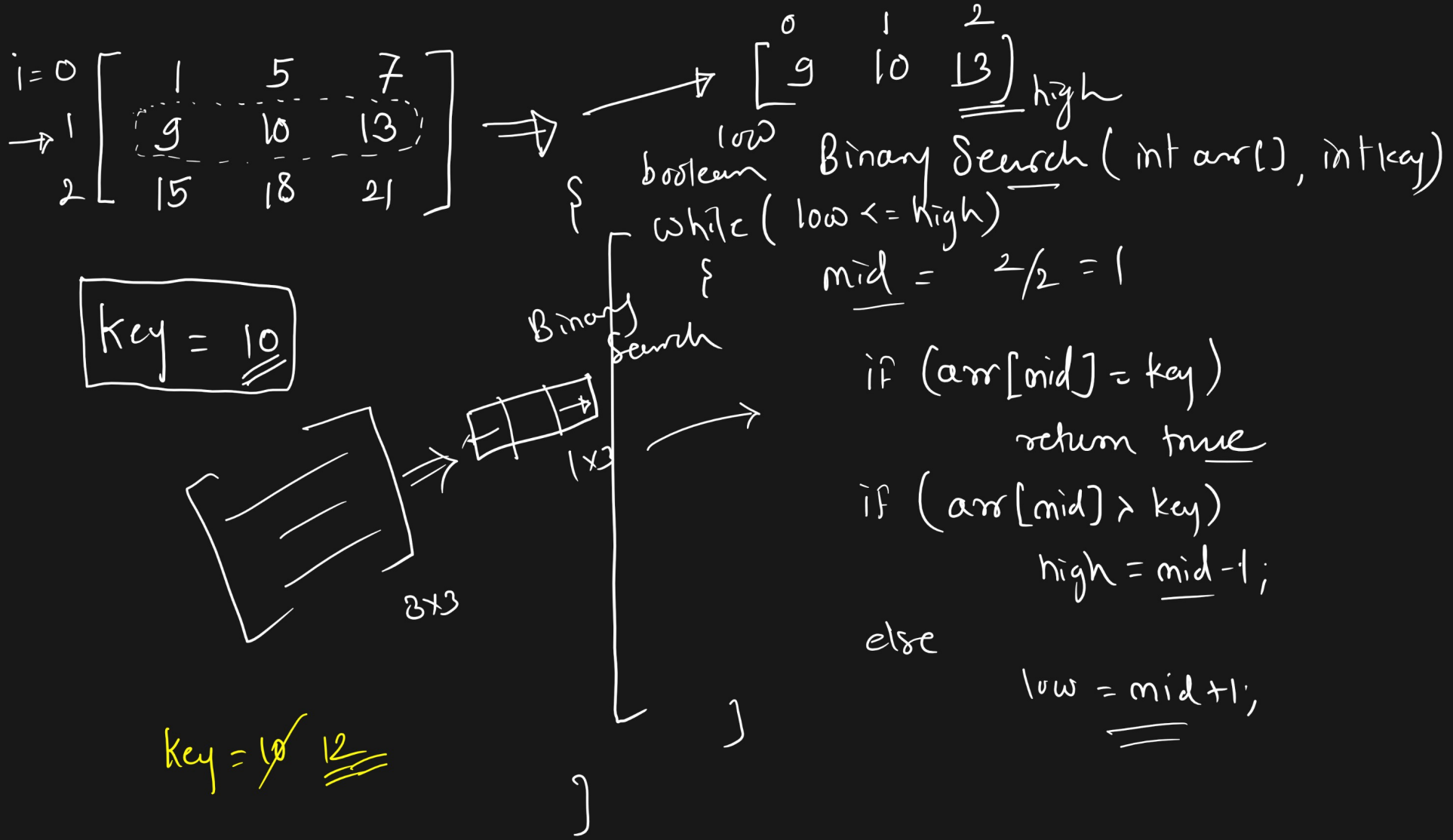
n^2

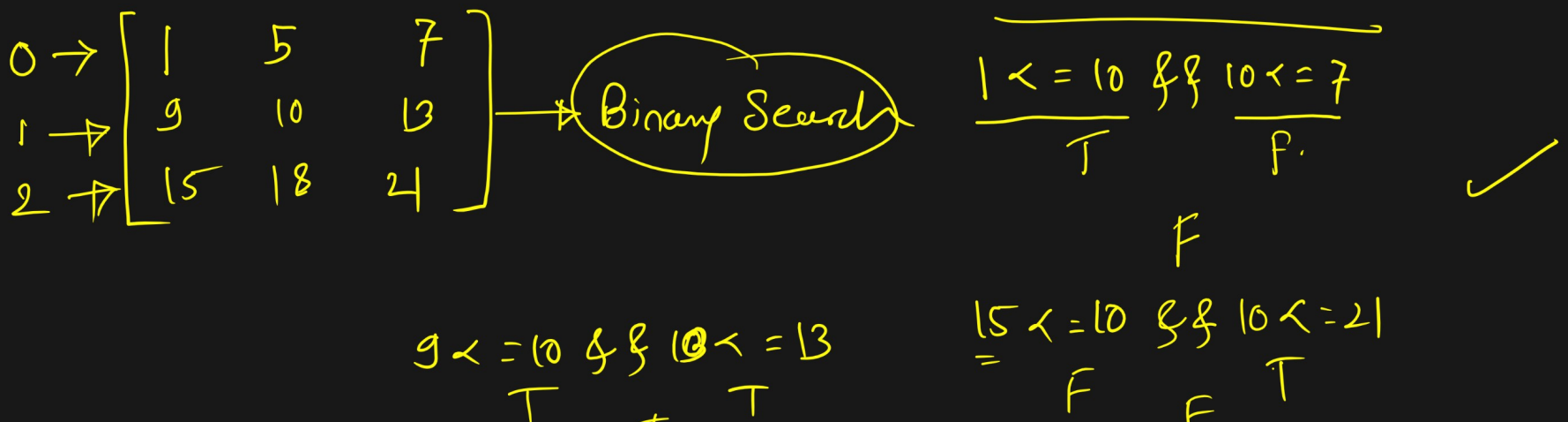
i × j = o(row×clm)

o(row×clm)

o(3×3)

j = 0 1 2

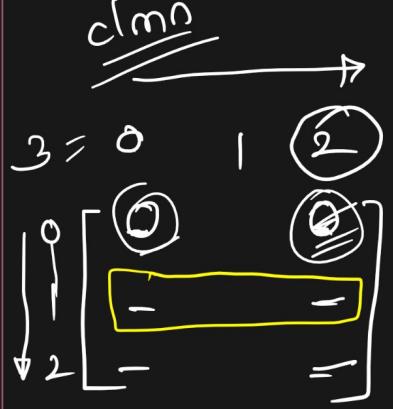

i=0
1
2
1 5 7
9 10 13
15 18 21
0 1 2
9 10 13
low
high
boolean Binary Search (int arr[], int key)
{
while (low <= high)
{
mid = 2/2 = 1
if (arr[mid] = key)
return true
if (arr[mid] > key)
high = mid-1;
else
low = mid+1;
}
}
Key = 10
Binary Search
1x3
3x3
Key = 10 12


0
1
2
1 5 7
9 10 13
15 18 21
Binary Search
1 <= 10 && 10 <= 7
T F
F
9 <= 10 && 10 <= 13
T T
15 <= 10 && 10 <= 21
F F T

How we can find out that ele

1) Find out row in that array be that ele will be present?
2) Find out ele using binary search on that row.

```
boolean ElementSearch ( int arr[][], int key)
{
    → int row = arr.length;
    → int clmn = arr[0].length;
      int rowIndex;
      for(int i=0; i< row; i++)
      {
          if (arr[i][0] <= key && key <= arr[i][clmn-1])
          {
              return binarySearch(arr[i], key);
          }
      }
```

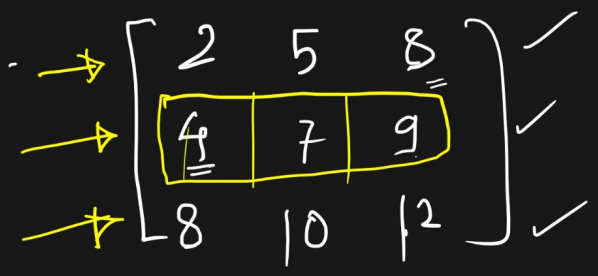


T/F

T/F → return false;
F

F

* key = 8

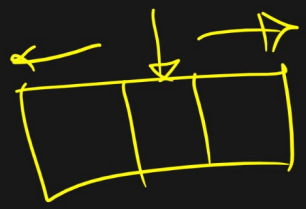



```
boolean EleSearch( int arr[][], int key)
{
    int row = arr.length;
    int clmn = arr[0].length;

    for( int i=0 ; i<row ; i++)
    {
        if ( binarySearch(arr[i], key) )
        {
            return true;
        }
    }
    return false;
```

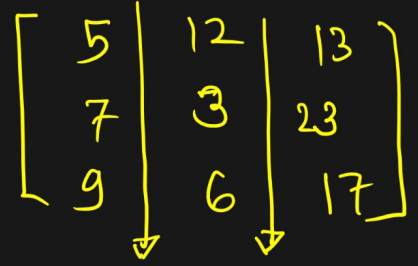
T/F

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<m; j++)
    {
        if ( _ == arr[i][j])
```



n log n

* H.W. Find out the column index that have max^{m} no of even no. & ——u—— that have max^{m} sum of ele.



String

→ What is String ⟹ Collection of element of char type

① Declr :- String s = "Hello Programmer";

JAVA → String s = new String ("Hello Programmer"); } Initialize

Python → s = "Hello Programmer"

② s = " _____ "

JAVA S.O.P (s)

Python print (s) } Print

o/p :
Hello Programmer

A) Display the each char of string

B) Reverse the string

↳ A) charwise

B) wordwise

A) i/p ⇒ Hello Programmer
remmargorP olleH } charwise

(B) :- o/p ⇒ Programmer Hello

Explaination :-

i/p s = " Hello Programmer"

o/p s = " Programmer Hello"

⇒ s = "Hello Programmer" ✓

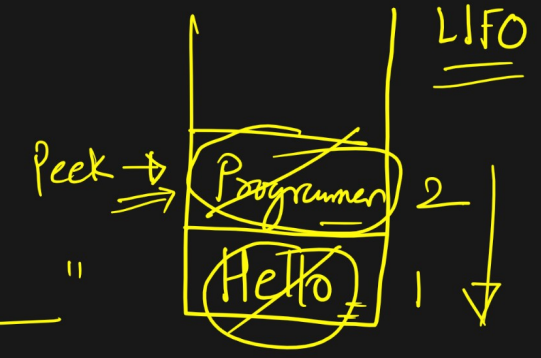
```
s = st" "; // "Hello Programmer "
Stack <String> st = new Stack <String>();
String str = "";
for ( int i = 0; i < s.length(); i++)
{
    char c = s.charAt(i);
    IF( c == ' ' )
    {
        st.push(str);
        str = "";
    }
    else
    {
        str = str + c;
```

Hello P

Str = Hello

Str = Programmer _

s = " _____ "



```
}
String reverse = "";
while( st.size() == 1)
{
    reverse = reverse + st.peek() + " ";   // reverse = Programmer -
    st.pop();
}
reverse = reverse + st.peek();   // reverse = P - Hello
st.pop();
S.o.p(reverse);
}
```

Stack :- → push() → enter elements ⟹ predefined

↓ → pop() → elements remove ⟹

Array / Linked

peek() → top element will be pointed by peek