Lec-25

20/06/25 8:19 PM

Given an integer `n`, return *the number of prime numbers that are strictly less than* `n`.

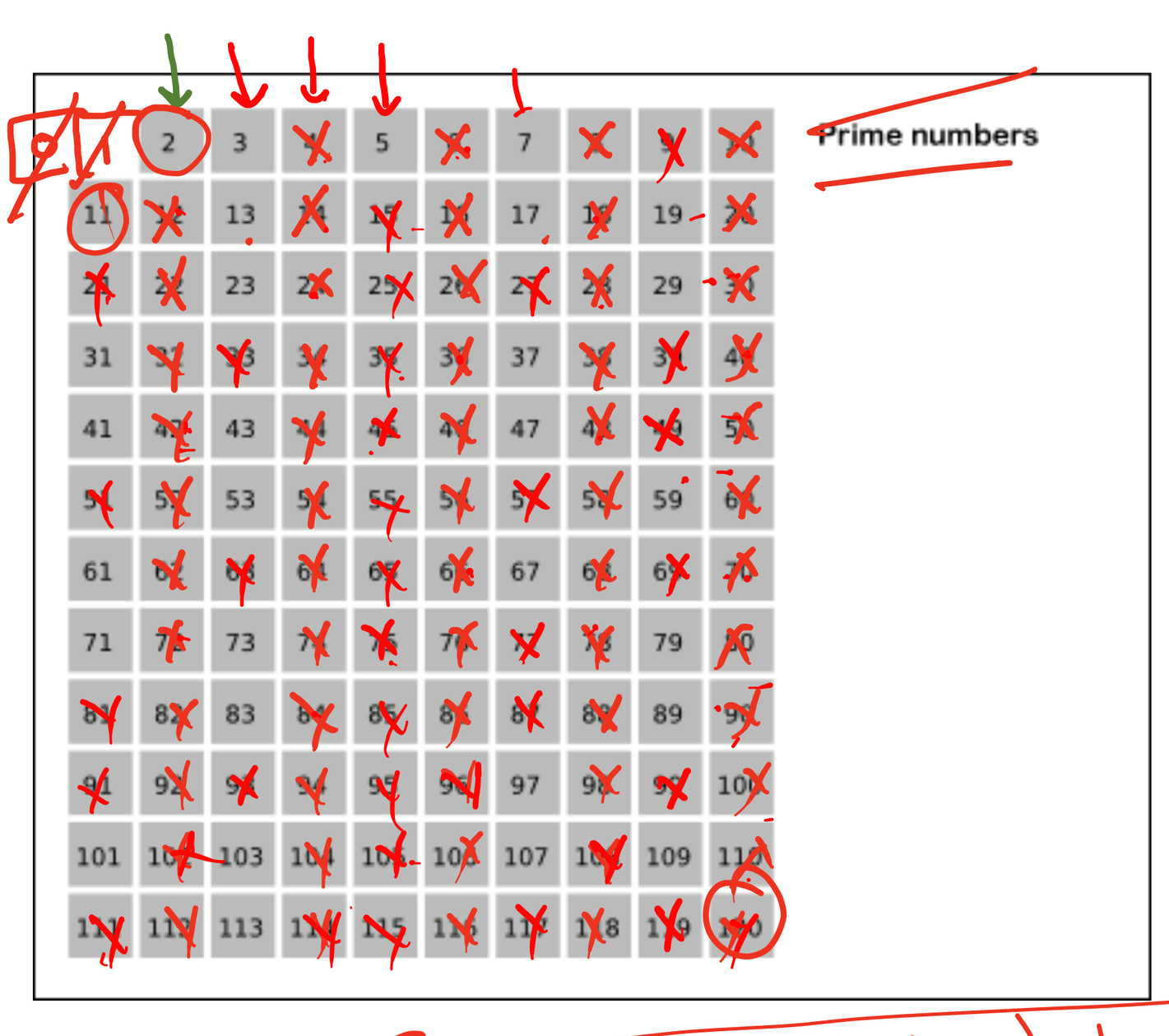
Example 1:

```
Input: n = 10
Output: 4
Explanation: There are 4 prime numbers less than 10, they
are 2, 3, 5, 7.
```

```java
public static void main(String[] args) {
        // TODO Auto-generated method stub
        int n = 9;
        System.out.println(isPrime(n));
    }

    public static boolean isPrime(int n) {
        int i = 2;
        while (i * i <= n) {
            if (n % i == 0) {
                return false;
            }
            i++;
        }
        return true;
    }
```

sieve of Eratosthenes



```java
int[] prime = new int[n];
        prime[0] = prime[1] = 1;// 0 & 1 is not a prime number
        for (int i = 2; i * i < prime.length; i++) {
            if (prime[i] == 0) {
                for (int j = 2; i * j < prime.length; j++) {
                    prime[i * j] = 1;
                }
            }
        }
```

```java
public static int[] Merge(int[] arr1, int[] arr2) {
        int n = arr1.length, m = arr2.length;
        int[] ans = new int[n + m];
        int i = 0, j = 0, k = 0;
        while (i < n && j < m) {
            if (arr1[i] < arr2[j]) {
                ans[k] = arr1[i];
                i++;
                k++;
            } else {
                ans[k] = arr2[j];
                j++;
                k++;
            }
        }
        // koi ek array rhgya
        while (i < n) {
            ans[k] = arr1[i];
            i++;
            k++;
        }
        while (j < m) {
            ans[k] = arr2[j];
            j++;
            k++;
        }
        return ans;
    }
```