

# SIEVE OF ERATOSTHENES

★ In this problem, we are given a number  $N$  and our job is to return an array containing all primes upto  $N$ .

Bruteforce solution is to obviously go from  $1 \rightarrow N$  and check whichever number is or is not prime.

Time Complexity here is  $O(N\sqrt{N})$  which is not at all feasible.

Optimal Solution is to create a boolean array of size  $N+1$

We take the first element, which represents 2 and iterate through the entire array and toggle all elements which is a multiple of 2 (as they can't be prime).

We do this iteration process only for elements which were marked as true.

Pseudocode :

```
allPrimeNumbers(N) {  
    prime[N+2] = []  
    for (i = 2 → N) {  
        prime[i] = True
```

```

    }
    for (i = 2 → √N) {
        if (prime[i] == True) {
            for (j = i2 → N, j += i) {
                prime[j] = 0
            }
        }
    }

```

```

    }
    ans = []
    for (i = 2 → N) {
        if (prime[i] == True) {
            ans.add(i)
        }
    }

```

```

    }
    return ans
}

```