

## prime number

소수에 직접적으로 연관된 문제는 많지 않지만 문제 풀이 도중 소수를 구할 필요가 있는 경우는 있는 편이다. 소수 판별 문제는 두 개 유형으로 나눌 수 있다. 첫번째는 어떤 수가 소수인지 판별하는 문제이고 두번째는 일정한 범위의 숫자들이 소수인지 판별하는 문제이다.

첫번째 경우에는 그 수의 제곱근까지만 확인해보면 된다.

```
1 bool primeNumber(int num){
2     for (int i = 1; i <= sqrt(num); i++)
3         if (num % i == 0) return false;
4     return true;
5 }
```

Listing 1: num의 소수 판별

이는  $O(\sqrt{n})$ 의 시간복잡도를 가진다.

두번째 경우에는 ‘에스토스테네스의 체’ 알고리즘을 사용한다.

```
1 void primeNumber(){
2     for (int i = 2; i <= sqrt(max); i++) {
3         if (num[i] == true)
4             for (int j = i + 1; j <= MAX; j += i)
5                 num[j] = false;
6     }
7 }
```

Listing 2: 2와 max 사이의 숫자들의 소수 판별

시간복잡도는  $O(n\log\log(n))$ 임이 알려져 있다.

## Greatest Common Divisor

최대공약수이다. 자주 나오지는 않지만 알아만 두자.  
최대공약수를 구할 때에는 ‘유클리드 호제법’을 사용한다.

```
1  int euclidean(int a, int b){  
2      while (true){  
3          int tmp = b;  
4          b = a % b;  
5          a = tmp;  
6          if (b == 0) break;  
7      }  
8      return a;  
9  }
```

Listing 3: a와 b의 최대공약수 구하기

## Data structure

## Sorting algorithm