

# Inclusion-Exclusion Principle

## CS 491 – Competitive Programming

Mattox Beckman

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  
DEPARTMENT OF COMPUTER SCIENCE

Spring, 2023

# Objectives

- ▶ Define the principle on inclusion / exclusion
- ▶ Use it to solve some actual problems

## Starting Example

- ▶ From Project Euler problem 1:  
*If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000*
- ▶ Try to find this out. Right now.
  - ▶ What did you have to do?

## Second Example

- ▶ Suppose your company has:
  - ▶ 20 Java programmers
  - ▶ 30 C programmers
  - ▶ 20 Python programmers
  - ▶ 15 know Java and C
  - ▶ 10 know C and Python
  - ▶ 10 know Java and Python
  - ▶ 5 know all three.
- ▶ How many programmers know at least one of these languages?
- ▶ If you have 70 employees total, how many know *none* of these languages?

# Primes

- ▶ From Geeks for Geeks:  
*Given  $N$  prime numbers and a number  $M$ , find out how many numbers from 1 to  $M$  are divisible by any of the  $N$  given prime numbers.*
- ▶ For this one, we need to make the powerset of the  $N$  primes. Use the bits of an integer: bit  $i$  set means we check for prime  $i$ .

## Setup Code

```
1  int count(int primes[], int m, int n)
2  {
3      int odd = 0, even = 0;
4      int counter, i, j, p = 1;
5      int pow_set_size = (1 << n);
6
7      for (counter = 1;
8           counter < pow_set_size;
9           counter++) {
10         p = 1;
11         for (j = 0; j < n; j++)    // check jth bit
12             if (counter & (1 << j))
13                 p *= primes[j];
```

## Continued

```
1      // popcount counts number of set bits
2      // if set bits is odd, then add to
3      if (__builtin_popcount(counter) & 1)
4          odd += (m / p);
5      else
6          even += (m / p);
7  }
8  return odd - even;
9  }
```