Competitive Programming

Science Honors Program (SHP)

Session 6

Christian Yongwhan Lim

Saturday, November 7, 2026

Overview

- Combinatorics
  - Finding Power of Factorial Divisor
  - Binomial Coefficients
  - Catalan Numbers
  - Inclusion-Exclusion Principle

Finding Power of Factorial Divisor

- You are given two numbers n and k.
- Find the largest power of k, x, such that $n!$ is divisible by k^x.

Solution (Prime k)

Thus, x equals

$$\left\lfloor \frac{n}{k} \right\rfloor + \left\lfloor \frac{n}{k^2} \right\rfloor + \ldots + \left\lfloor \frac{n}{k^i} \right\rfloor + \ldots$$

Implementation

```
int fact_pow (int n, int k) {
    int res = 0;
    while (n) {
        n /= k;
        res += n;
    }
    return res;
}
```

Solution (Composite k)

Define a_i to be the number of times prime k_i divides evenly into n!, where k_i through k_m are the unique prime factors of k and k is expressed as follows:

$$k = k_1^{p_1} \cdot \ldots \cdot k_m^{p_m}$$

Then, the number of times k divides into n! is:

$$\min_{i=1\ldots m} \frac{a_i}{p_i}$$

Binomial Coefficients: Formulas

$$(a+b)^n = \binom{n}{0}a^n + \binom{n}{1}a^{n-1}b + \binom{n}{2}a^{n-2}b^2 + \cdots + \binom{n}{k}a^{n-k}b^k + \cdots + \binom{n}{n}b^n$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Binomial Coefficients: Properties

$$\binom{n}{k} = \binom{n}{n-k}$$

$$\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$$

Binomial Coefficients: Properties

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

$$\sum_{m=0}^{n} \binom{m}{k} = \binom{n+1}{k+1}$$

Binomial Coefficients: Properties

$$\sum_{k=0}^{m} \binom{n+k}{k} = \binom{n+m+1}{m}$$

$$\binom{n}{0}^2 + \binom{n}{1}^2 + \cdots + \binom{n}{n}^2 = \binom{2n}{n}$$

Binomial Coefficients: Properties

$$1\binom{n}{1} + 2\binom{n}{2} + \cdots + n\binom{n}{n} = n2^{n-1}$$

$$\binom{n}{0} + \binom{n-1}{1} + \cdots + \binom{n-k}{k} + \cdots + \binom{0}{n} = F_{n+1}$$

Binomial Coefficients: Naive Implementation

```
int C(int n, int k) {
    int res = 1;
    for (int i = n - k + 1; i <= n; ++i)
        res *= i;
    for (int i = 2; i <= k; ++i)
        res /= i;
    return res;
}
```

Binomial Coefficients: Improved Implementation

```
int C(int n, int k) {
    double res = 1;
    for (int i = 1; i <= k; ++i)
        res = res * (n - k + i) / i;
    return (int)(res + 0.01);
}
```

Pascal's Triangle: Implementation

```
const int maxn = ...;
int C[maxn + 1][maxn + 1];
C[0][0] = 1;
for (int n = 1; n <= maxn; ++n) {
    C[n][0] = C[n][n] = 1;
    for (int k = 1; k < n; ++k)
        C[n][k] = C[n - 1][k - 1] + C[n - 1][k];
}
```

Catalan Numbers: Motivating Examples

- Number of correct bracket sequence consisting of n opening and n closing brackets.
- The number of rooted full binary trees with $n + 1$ leaves (vertices are not numbered). A rooted binary tree is full if every vertex has either two children or no children.
- The number of ways to completely parenthesize $n + 1$ factors.
- The number of ways to connect the $2n$ points on a circle to form n disjoint chords.
- The number of non-isomorphic full binary trees with n internal nodes (i.e. nodes having at least one son).
- ...

Catalan Numbers

- 1
- 1
- 2
- 5
- 14
- 42
- 132
- 429
- 1430
- ...

Catalan Numbers: Recursive Formula

$$C_0 = C_1 = 1$$

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k}, n \geq 2$$

Catalan Numbers: Implementation

```
int catalan[MAX];
void init() {
    catalan[0] = catalan[1] = 1;
    for (int i=2; i<=n; i++) {
        catalan[i] = 0;
        for (int j=0; j < i; j++) {
            catalan[i] += (catalan[j] *
                           catalan[i-j-1]) % MOD;
            if (catalan[i] >= MOD) catalan[i] -= MOD;
        }
    }
}
```

Catalan Numbers: Analytical Formula

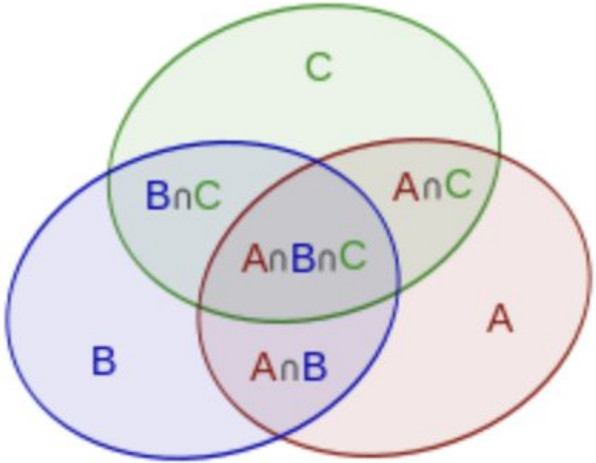
$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

$$C_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}, n \geq 0$$

The Inclusion-Exclusion Principle

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq J \subseteq \{1,2,\ldots,n\}} (-1)^{|J|-1} \left|\bigcap_{j \in J} A_j\right|$$

The Inclusion-Exclusion Principle: Venn Diagrams



$$S(A \cup B \cup C) = S(A) + S(B) + S(C) - S(A \cap B) - S(A \cap C) - S(B \cap C) + S(A \cap B \cap C)$$

The number of relative primes in a given interval

- Given two numbers `n` and `r`, count the number of integers in the interval `[1, r]` that are relatively prime to `n` (their greatest common divisor is `1`).

The Main Idea

- We will denote the prime factors of n as p_i (i = 1, ..., k).

- How many numbers in the interval [1, r] are divisible by p_i?

Solution Sketch

- The answer to this question is: r/p_i.

- However, if we simply sum these numbers, some numbers will be counted several times (those that share multiple p_i as their factors).

- Therefore, it is necessary to use the inclusion-exclusion principle.
- We will iterate over all 2^k subsets of p_i's, calculate their product and add or subtract the number of multiples of their product.

Implementation

```
int solve (int n, int r) {
    vector<int> p;
    for (int i=2; i*i<=n; ++i)
        if (n % i == 0) {
            p.push_back (i);
            while (n % i == 0)
                n /= i;
        }
    if (n > 1)
        p.push_back (n);
```

Implementation

```
    int sum = 0;
    for (int msk=1; msk<(1<<p.size()); ++msk) {
        int mult = 1, bits = 0;
        for (int i=0; i<(int)p.size(); ++i)
            if (msk & (1<<i)) ++bits, mult *= p[i];
        int cur = r / mult;
        if (bits % 2 == 1) sum += cur;
        else sum -= cur;
    }
    return r - sum;
}
```


BREAK

Group Exercise

- You may work in groups with those sitting near you.
- Each person must submit their own solutions.
- You have **75 minutes**.
- Afterward, you will present your solution!


THANK YOU