수학 2

최백준 choi@startlink.io

a^b

a^b

a^b

- a의 b제곱을 빠르게 구해야 한다.

```
int ans = 1;
for (int i=1; i<=b; i++) {
    ans = ans * a;
}
```

- 직관적인 방법이지만 O(b)라는 시간이 걸리게 된다.
- 따라서, 조금 더 빠른 방법이 필요하다.

a^b

분할정복 이용하기

- 분할정복을 이용해서 구할 수 있다.
- $a^{2b} = a^b \times a^b$
- $a^{2b+1} = a \times a^{2b}$

a^b

분할정복 이용하기

```
int calc(int a, int b) {
    if (b == 0) {
        return 1;
    } else if (b == 1) {
        return a;
    } else if (b % 2 == 0) {
        int temp = calc(a, b/2);
        return temp * temp;
    } else { // b % 2 == 1
        return a * calc(a, b-1);
    }
}
```

a^b

분할정복 이용하기

- 이 부분을

```
} else if (b % 2 == 0) {
    int temp = calc(a, b/2);
    return temp * temp;
}
```

- 아래와 같이 구현 하면 O(b) 이다. 하지만, GCC 컴파일러 최적화로 매우 빠르게 수행된다.

```
} else if (b % 2 == 0) {
    return calc(a, b/2) * calc(a, b/2);
}
```

a^b

분할정복 이용하기

- 이진수의 원리를 이용해서도 구할 수 있다.

```
int calc(int a, int b) {
    int ans = 1;
    while (b > 0) {
        if (b % 2 == 1) {
            ans *= a;
        }
        a = a * a;
        b /= 2;
    }
    return ans;
}
```

a^b

분할정복 이용하기

- 예를 들어, 3의 27 제곱인 경우를 생각해보자.
- 27은 이진수로 11011 이다.
- $27 = 2^0 + 2^1 + 2^3 + 2^4$
- $27 = 1 + 2 + 8 + 16$
- $3^{27} = 3^{1+2+8+16}$
- $3^{27} = 3^1 \times 3^2 \times 3^8 \times 3^{16}$
- 을 이용해서 a를 계속해서 a×a로 곱해가면서 제곱을 구하게 된다.

곱셈

https://www.acmicpc.net/problem/1629

- 자연수 A를 B번 곱한 수를 C로 나눈 나머지를 구하는 문제

곱셈

https://www.acmicpc.net/problem/1629

- 분할 정복: http://codeplus.codes/84a3c5ffa0b442f9ada06fb799f32689
- 이진수 응용: http://codeplus.codes/8e5d1ece6a9c457585307978d21dc26e

행렬

행렬 덧셈

https://www.acmicpc.net/problem/2738

- 두 행렬을 입력받고 덧셈을 수행하는 문제

```
for (int i=0; i<n; i++) {
    for (int j=0; j<m; j++) {
        c[i][j] = a[i][j]+b[i][j];
    }
}
```

행렬 곱셈

https://www.acmicpc.net/problem/2740

- 두 행렬을 입력받고 곱셈을 수행하는 문제

```
for (int i=0; i<n; i++) {
    for (int j=0; j<r; j++) {
        c[i][j] = 0;
        for (int k=0; k<m; k++) {
            c[i][j] += a[i][k]*b[k][j];
        }
    }
}
```

행렬 제곱

https://www.acmicpc.net/problem/10830

- 행렬 A의 B제곱을 구하는 문제

행렬 제곱

https://www.acmicpc.net/problem/10830

- 소스: http://codeplus.codes/6676ab4abbf64b21845fd3a7b7232596

피보나치 수

피보나치 수

Fibonacci Number

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$
- 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ⋯

피보나치 수

https://www.acmicpc.net/problem/2747

- N번째 피보나치 수를 구하는 문제 (N ≤ 45)
- 소스: http://codeplus.codes/d2df6ce689f543099ea7dc3ce91ddbc4

피보나치 수 2

https://www.acmicpc.net/problem/2748

- N번째 피보나치 수를 구하는 문제 ($N \le 90$)
- 90번째 피보나치 수는 int 범위를 넘어간다.
- 소스: http://codeplus.codes/add8ab25102a427dbbde39c3360a6b82

피사노 주기

Pisano Period

- 피보나치 수를 K로 나눈 나머지는 주기를 갖는다.
- 이것을 피사노 주기라고 한다.
- 3으로 나누었을 때의 주기는 8이다.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F_n	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610
F_n%3	0	1	1	2	0	2	2	1	0	1	1	2	0	2	2	1

피보나치 수 3

https://www.acmicpc.net/problem/2749

- N번째 피보나치 수를 M = 1,000,000으로 나눈 나머지를 구하는 문제
- N ≤ 1,000,000,000,000,000,000
- 피사노 주기를 이용해서 주기를 찾고 문제를 풀 수 있다.
- 주기의 길이가 K이면
- N번째 피보나치 수를 M으로 나눈 나머지는 N%K 번째 피보나치 수와 같다.
- $M = 10^k$ 일 때, $k > 2$ 라면, 주기는 항상 $15 \times 10^{k-1}$ 이다.
- 이 사실을 모른다고 해도, 주기를 구하는 코드를 이용해서 정답을 구할 수 있다.

피보나치 수 3

https://www.acmicpc.net/problem/2749

- 소스: http://codeplus.codes/6be782671077479e9338af0df75b0bf6

피보나치 수 6

https://www.acmicpc.net/problem/11444

- N번째 피보나치 수를 M = 1,000,000,007으로 나눈 나머지를 구하는 문제
- N ≤ 1,000,000,000,000,000,000
- 주기가 어떻게 될 지 알 수 없다.

피보나치 수

Fibonacci Number

- $\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$
- $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$

피보나치 수

Fibonacci Number

- $F_{2n-1} = F_n^2 + F_{n-1}^2$
- $F_{2n} = (F_{n-1} + F_{n+1})F_n = (2F_{n-1} + F_n)F_n$

피보나치 수 6

https://www.acmicpc.net/problem/11444

- 행렬 제곱: http://codeplus.codes/3473abb26dfd407e940faaf5eb823bab
- 분할 정복: http://codeplus.codes/4b782640c1d04cf9b9e719ed07e727d9

이항 계수

이항 계수

Binomial Coefficient

- n개중에 k개를 순서 없이 고르는 방법 nCm
- $\binom{n}{k}$로 쓴다.
- $\frac{n!}{k!(n-k)!}$ 이다.
- $\frac{n \times (n-1)\times \ldots(n-k+1)}{k!}$ 와 같다.
- 구해보자!

이항 계수 1

https://www.acmicpc.net/problem/11050

- $\binom{n}{k}$ 를 구하는 문제
- $1 \leq n \leq 10$, $0 \leq k \leq n$ 이기 때문에
- $\frac{n!}{k!(n-k)!}$ 값을 구하면 된다.
- 시간 복잡도: O(N)

이항 계수 1

https://www.acmicpc.net/problem/11050

- 소스: http://codeplus.codes/9d82655b994a46248735110089275 2c3

파스칼의 삼각형

Pascal's Triangle

- 이항 계수를 삼각형 모양으로 배열
- n번 줄에는 수를 n개만 쓴다.
- 각 줄의 첫 번째와 마지막 수는 1이다.
- 나머지 수는 윗 줄의 왼쪽 수와 오른쪽 수를 더해서 만든다.

파스칼의 삼각형

Pascal's Triangle

- 5까지 파스칼의 삼각형

```
                  1
               1     1
            1     2     1
         1     3     3     1
      1     4     6     4     1
   1     5    10    10     5     1
1     6    15    20    15     6     1
```

파스칼의 삼각형

Pascal's Triangle

- C[n][k] = n번줄의 k번째 수라고 했을 때
- C[n][1] = 1, C[n][n] = 1
- C[n][k] = C[n-1][k-1] + C[n-1][k]
- 로 정의할 수 있음
- C[n][k]는 $\binom{n}{k}$ 이다.

파스칼의 삼각형

Pascal's Triangle

- C[n][k] = C[n-1][k-1] + C[n-1][k]
- C[n][k] 는 $\binom{n}{k}$ 를 나타내기 때문에, n개 중에 k개를 순서 없이 고르는 방법이다.
- n개 중에 k개를 순서 없이 고른다면 다음과 같은 두 가지 경우가 가능하다

1. n번째를 고른 경우
2. n번째를 고르지 않은 경우

파스칼의 삼각형

Pascal's Triangle

- C[n][k] = C[n-1][k-1] + C[n-1][k]
- C[n][k] 는 $\binom{n}{k}$ 를 나타내기 때문에, n개 중에 k개를 순서 없이 고르는 방법이다.
- n개 중에 k개를 순서 없이 고른다면 다음과 같은 두 가지 경우가 가능하다

1. n번째를 고른 경우
   - n번째를 골랐기 때문에, n-1개 중에 k-1개를 골랐어야 한다. C[n-1][k-1]
2. n번째를 고르지 않은 경우
   - n번째를 고르지 않았기 때문에, n-1개 중에 k개를 골랐어야 한다. C[n-1][k]

이항 계수 2

https://www.acmicpc.net/problem/11051

- $\binom{n}{k}$ 을 10,007로 나눈 나머지를 구하는 문제
- $1 \le n \le 1{,}000$, $0 \le k \le n$ 이기 때문에
- 파스칼의 삼각형을 이용해서 구할 수 있다.
- 시간 복잡도: O(NK)

이항 계수 2

https://www.acmicpc.net/problem/11051

- 소스: http://codeplus.codes/082cc46c9d504a8f88626f10a5977ae6

이항 계수

Binomial Coefficient

- $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$
- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ $(1 \leq \text{k} \leq \text{n-1})$
- $\binom{n}{k} = \binom{n}{n-k}$

이항 계수

Binomial Coefficient

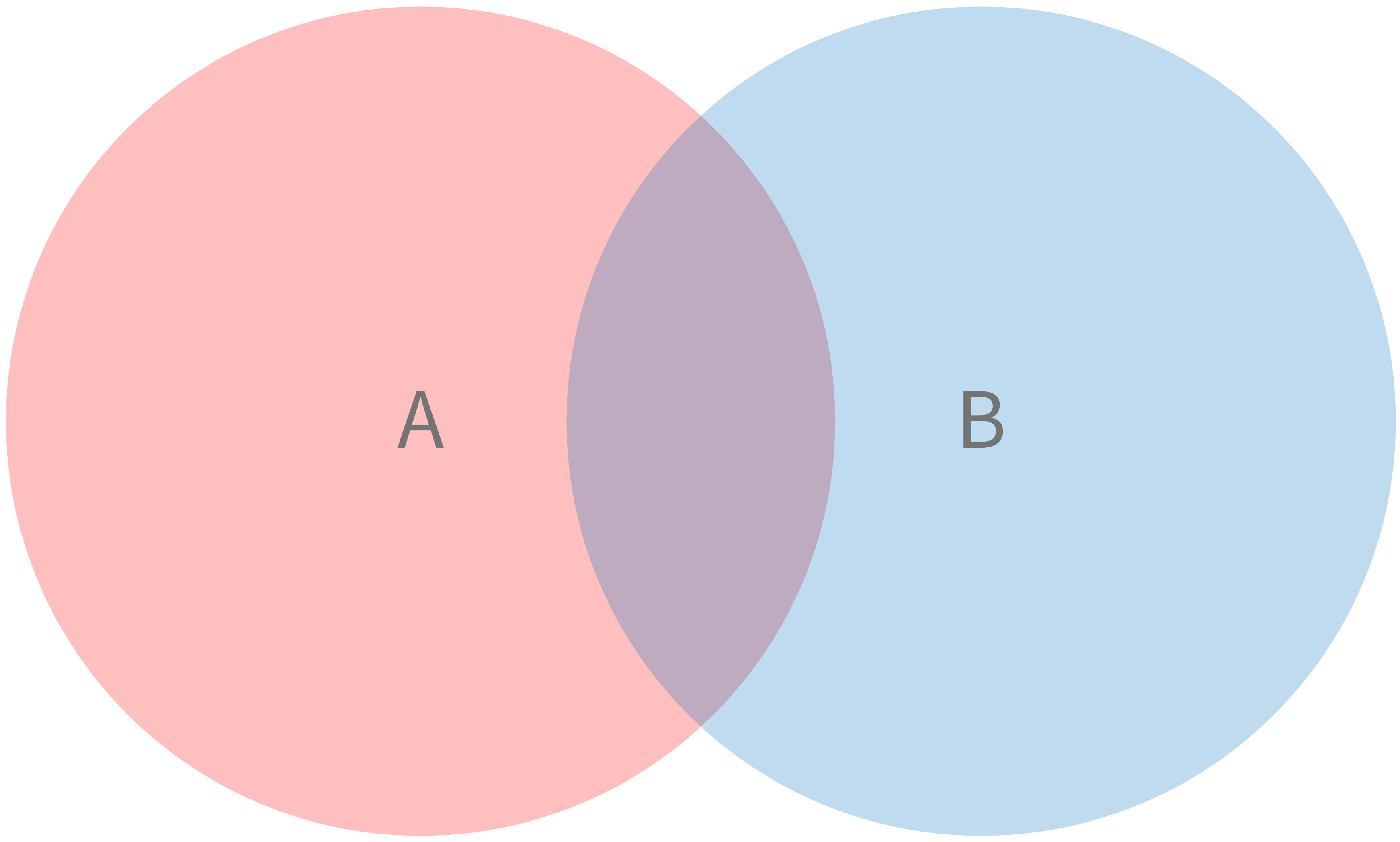
- n개 중에 k개를 중복 없이 뽑는 방법의 수 $\binom{n}{k}$
- n개 중에 k개를 중복을 허용하면서 뽑는 방법의 수 $\binom{n+k-1}{k}$
- 0과 1로만 이루어진 문자열의 개수 $\binom{n+k}{k}$
- 0과 1로만 이루어진 문자열의 개수 (1은 연속하지 않음) $\binom{n+1}{k}$
- 카탈란 수 $\frac{1}{n+1}\binom{2n}{n}$

포함-배제의 원리

포함-배제의 원리

Inclusion-exclusion Principle

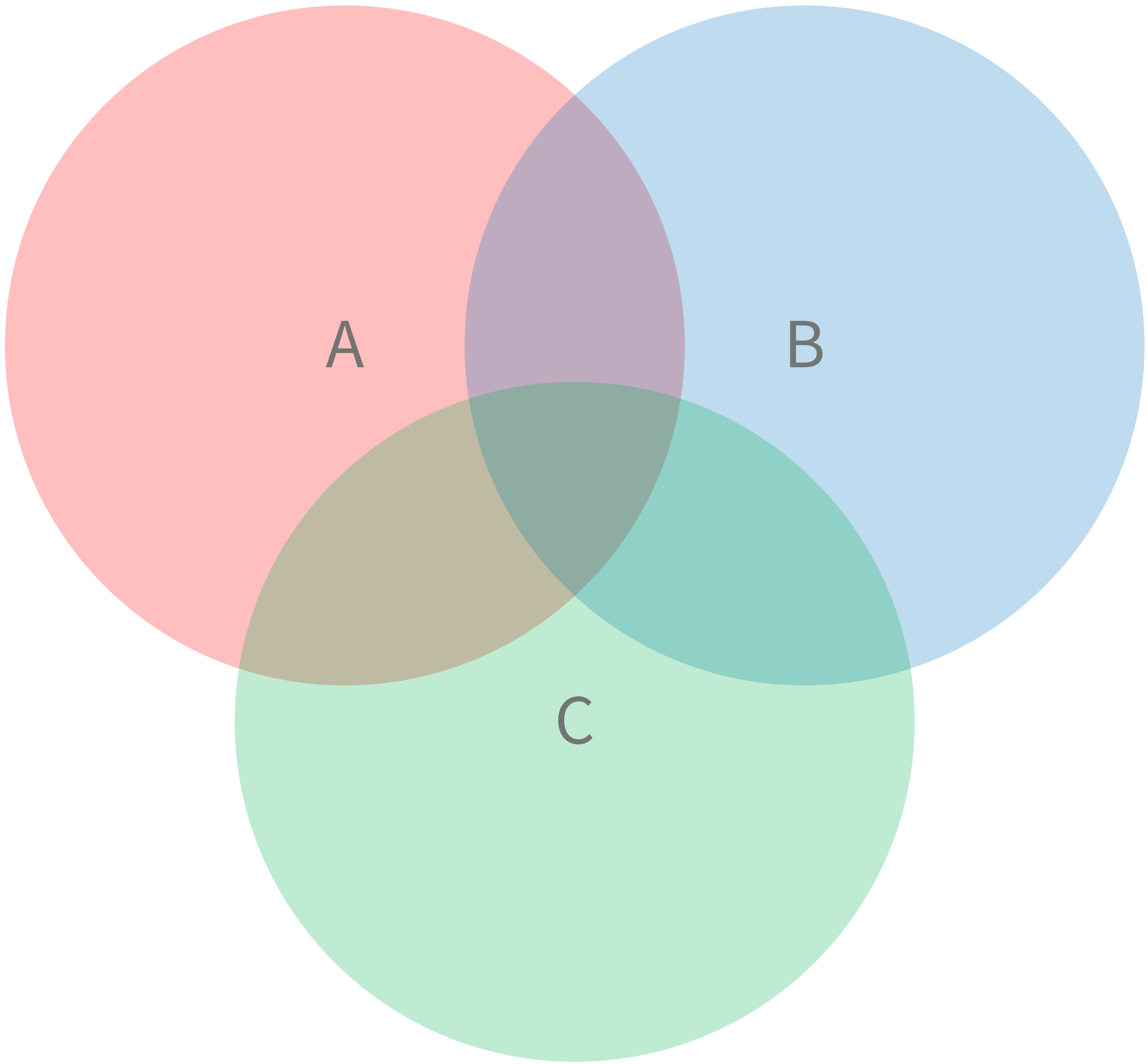
- 집합의 교집합 크기를 구할때 사용하는 방법
- $|A \cup B| = |A| + |B| - |A \cap B|$



포함-배제의 원리

Inclusion-exclusion Principle

- 집합의 교집합 크기를 구할때 사용하는 방법
- $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$



포함-배제의 원리

Inclusion-exclusion Principle

- N개의 집합의 교집합이라고 한다면, 다음과 같이 일반화할 수 있다.
- 집합의 크기를 포함
- 두 집합의 교집합 크기를 제외
- 세 집합의 교집합 크기를 포함
- 네 집합의 교집합 크기를 제외
- …

소수의 배수

https://www.acmicpc.net/problem/17436

- N개의 소수와 자연수 M이 주어졌을 때
- M 이하의 자연수 중에서 N개의 소수 중 적어도 하나로 나누어 떨어지는 수의 개수를 세는 문제
- $1 \leq N \leq 10$
- $1 \leq M \leq 10^{12}$
- 1 ≤ 소수 ≤ 100

소수의 배수

https://www.acmicpc.net/problem/17436

- 100이하의 자연수 중 2와 3으로 나누어 떨어지는 수의 개수를 센다면
- |2의 배수| + |3의 배수| - |2×3의 배수|
- 100이하의 자연수 중 2, 3, 5로 나누어 떨어지는 수의 개수를 센다면
- |2의 배수| + |3의 배수| + |5의 배수| - |2×3의 배수| - |2×5의 배수| - |3×5의 배수| + |2×3×5의 배수|

소수의 배수

https://www.acmicpc.net/problem/17436

- 포함 배제를 이용해서 구현할 수 있다.

소수의 배수

https://www.acmicpc.net/problem/17436

```
for (int i=1; i<(1<<n); i++) {
    int cnt=0; long long p=1;
    for (int j=0; j<n; j++) {
        if (i&(1<<j)) {
            p *= a[j]; cnt += 1;
        }
    }
    if (cnt % 2 == 0) {
        ans -= (m/p);
    } else {
        ans += (m/p);
    }
}
```

소수의 배수

https://www.acmicpc.net/problem/17436

- 소스: http://codeplus.codes/283d272f33bf4e4e8aeeb91a243f69fb

소수의 배수

https://www.acmicpc.net/problem/17436

```
long long go(int index, long long num) {
    if (index >= n) return 0;
    long long ans = 0;
    if (num*a[index] > m) return 0;
    ans += m/(num*a[index]);
    ans += go(index+1, num);
    ans -= go(index+1, num*a[index]);
    return ans;
}
```

소수의 배수

https://www.acmicpc.net/problem/17436

- 소스: http://codeplus.codes/58f364608fd44083ba4681429a8dd0ac