Kosen Club CP 勉強会

MATH

1.GCD/LCM/ExtGCD O(logn)

```
Int gcd(Int a, Int b) {
  return b != 0 ? gcd(b, a % b) : a;
}
Int lcm(Int a, Int b) {
  return a * b / gcd(a, b);
}
// a x + b y = gcd(a, b)
Int extgcd(Int a, Int b, Int &x, Int &y) {
  Int g = a; x = 1; y = 0;
  if (b != 0) g = extgcd(b, a % b, y, x), y -= (a / b) * x;
  return g;
}
```

2.Fast Power O(logk)

```
Int powMod(Int x, Int k, Int m) {
  if (k == 0)     return 1;
  if (k % 2 == 0) return powMod(x*x % m, k/2, m);
  else            return x*powMod(x, k-1, m) % m;
}
```

3.Is Prime? O(Sqrt(n))

```
bool is_prime(long long N) {
    if (N == 1) return false;
    for (long long i = 2; i * i <= N; ++i) {
        if (N % i == 0) return false;
    }
    return true;
}
```

4.All divisors O(Sqrt(n))

```
vector<long long> enum_divisors(long long N) {
   vector<long long> res;
   for (long long i = 1; i * i <= N; ++i) {
      if (N % i == 0) {
         res.push_back(i);
      if (N/i != i) res.push_back(N/i);
      }
   }
   sort(res.begin(), res.end());
   return res;
}
```

5.Prime Decomposition O(Sqrt(n))

```
vector<pair<long long, long long> > res;
for (long long a = 2; a * a <= N; ++a) {
    if (N % a != 0) continue;
    long long ex = 0;
   while (N % a == 0) {
        ++ex;
        N /= a;
    }
    res.push_back({a, ex});
}
```

6.Mod Operations O(1) – O(logn)

Basic operations:

+:

(A + B) % MOD

-:

(A + MOD – B) % MOD

*:

(A * B) % MOD

/:

(A * powMod(B, MOD-2, MOD)) % MOD

```
Int powMod(Int x, Int k, Int m) {
  if (k == 0)     return 1;
  if (k % 2 == 0) return powMod(x*x % m, k/2, m);
  else            return x*powMod(x, k-1, m) % m;
}
```

7.fact, inverse fact, inverse O(n)

```
Int fact[N], invfact[N], inv[N];

void init()

{

    fact[0] = 1;

    for (int i = 1; i < N; i++) {

        fact[i] = fact[i - 1] * i % MOD;

    }

    inv[1] = 1;

    for (int i = 2; i < N; i++) {

        inv[i] = -inv[MOD%i] * (MOD / i) % MOD;

        if (inv[i] < 0) inv[i] += MOD;

    }

    invfact[0] = 1;

    for (int i = 1; i < N; i++) {

        invfact[i] = invfact[i - 1] * inv[i] % MOD;

    }

}
```

```
Int nCk(Int n, Int k) // (n!/k! * (n-k)!)

{

    return fact[n] * invfact[k] % MOD *
            invfact[n - k] % MOD;

}

Int nPk(Int n, Int k) // (n!/(n-k)!)

{

    return fact[n] * invfact[n - k] % MOD;

}
```

References

Euclid: AtCoder 版！マスター・オブ・整数 (最大公約数編) - Qiita

Primes: AtCoder 版！マスター・オブ・整数 (素因数分解編) – Qiita

Mod:「1000000007 で割ったあまり」の求め方を総特集！ ～ 逆元から離散対数まで ～ - Qiita