りーぜんと

テンプレ

ソースコード 1　テンプレ

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
const int INF = 1e18;
const int MOD = 1e9 + 7;
```

ソースコード 2　ループ

```
#define _overload(_1, _2, _3, name, ...) name
#define _rep(i, n) repi(i, 0, n)
#define repi(i, a, b) for(int i = (int)(a); i < (int)(b); i++)
#define rep(...) _overload(__VA_ARGS__, repi, _rep)(__VA_ARGS__)
#define _rev(i, n) revi(i, n, 0)
#define revi(i, a, b) for(int i = (int)(a - 1); i >= (int)(b); i--)
#define rev(...) _overload(__VA_ARGS__, revi, _rev)(__VA_ARGS__)
#define each(i, n) for(auto&& i: n)
```

データ構造

ソースコード 3　UnionFind

```
struct UnionFind {
    vector<int> t;
    UnionFind(int size): t(size, -1) {}
    int root(int x) {
        return t[x] < 0 ? x : t[x] = root(t[x]);
    }
    int size(int x) {
        return -t[root(x)];
    }
    bool isSame(int x, int y) {
        return root(x) == root(y);
    }
    bool unite(int x, int y) {
        x = root(x), y = root(y);
        if(x != y) {
            if(t[y] < t[x]) swap(x, y);
            t[x] += t[y], t[y] = x;
        }
```

```
        return x != y;
    }
};
```

ソースコード 4　セグ木

```
struct SegmentTree {
  int n = 1;
  vector<int> t;
  SegmentTree(vector<int> v) {
      while(n < v.size()) n *= 2;
      t.resize(2 * n - 1, INF);
      rep(i, v.size()) t[i + n - 1] = v[i];
      rev(i, n - 1) t[i] = min(t[2 * i + 1], t[2 * i + 2]); // 区間最小
  }
  void update(int x, int val) {
      x += n - 1;
      t[x] = val;
      while(x > 0){
          x--; x /= 2;
          t[x] = min(t[2 * x + 1], t[2 * x + 2]); // 区間最小
      }
  }
  int query(int a, int b, int now = 0, int l = 0, int r = -1) {
      if(r < 0) r = n;
      if(r <= a || b <= l) return INF; // の単位元 min
      if(a <= l && r <= b) return t[now];
      int c1 = query(a, b, 2 * now + 1, l, (l + r) / 2);
      int c2 = query(a, b, 2 * now + 2, (l + r) / 2, r);
      return min(c1, c2);
  }
};
```

ソースコード 5　回文

```
// 最長回文v[iが]文字目中心の差最大値 i
vector<int> manacher(string &s) {
    vector<int> radius(s.size());
    int i = 0, j = 0;
    while(i < s.size()) {
        while(i - j >= 0 && i + j < s.size() && s[i - j] == s[i + j]) {
            ++j;
        }
        radius[i] = j;
        int k = 1;
        while(i - k >= 0 && i + k < s.size() && k + radius[i - k] < j) {
            radius[i + k] = radius[i - k];
```

```
            ++k;
        }
        i += k;
        j -= k;
    }
    return radius;
}
```

数学

ソースコード 6　素因数分解

```
map<int, int> factorization(int n) {
    map<int, int> res;
    rep(i, 2, sqrt(n) + 1) {
        if(i > 3) i++;
        while(!(n%i)) {
            res[i]++;
            n /= i;
        }
    }
    if(n > 1) {
        res[n]++;
    }
    return res;
}
```

ソースコード 7　約数列挙

```
vector<int> divisor(int n) {
    vector<int> res;
    rep(i, 1, sqrt(n) + 1) {
        if(!(n % i)) {
            res.push_back(i);
            if(i * i < n) res.push_back(n / i);
        }
    }
    return res;
}
```

ソースコード 8　最小公倍数

```
int lcm(int a, int b) {return a / gcd(a, b) * b;}
```

ソースコード 9　累乗

```
int factorial(int a) {return a < 2 ? 1 : factorial(a - 1) * a;}
```

ソースコード 10　総和

```
int summation(int a) {return a < 1 ? 0 : (a * a + a) / 2;}
```

ソースコード 11　組み合わせ

```
int comb(int n, int r) {
  int res = 1;
  rep(i, 1, r + 1) {
    res *= n--;
    res /= i;
  }
  return res;
}
```

ソースコード 12　素数判定

```
bool isPrime(int n) {
  rep(i, 2, sqrt(n) + 1) {
    if(i > 3) i++;
    if(!(n % i)) return false;
  }
  return true;
}
```

bound

ソースコード 13　bound

```
template<class T>
int lb(vector<T>& x,T n){return lower_bound(all(x) , n) - x.begin();}
template<class T>
int ub(vector<T>& x,T n){return upper_bound(all(x) , n) - x.begin();}
```

MOD

ソースコード 14　二分累乗とか

```
int extgcd(int a, int b, int &x, int &y) {
  int g = a;
  x = 1, y = 0;
  if(b) {
    g = extgcd(b, a % b, y, x);
    y -= a / b * x;
  }
  return g;
}
```

```
10
11 int invmod(int a, int m = MOD) {
12   int x = 0, y = 0;
13   extgcd(a, m, x, y);
14   return (x + m) % m;
15 }
16
17 int modpow(int a, int n){
18   if(n < 1) return 1;
19   return modpow(a * a % MOD, n / 2) * ((n % 2) ? a : 1) % MOD;
20 }
21
22 int modfact(int a){
23   if(a < 2) return 1;
24   return a * modfact(a - 1) % MOD;
25 }
```

やの

テンプレ

ソースコード 15　テンプレ

```
1 #include<iostream>
2 #include<vector>
3 #include<algorithm>
4 #include<string>
5 #include<map>
6 #include<set>
7 #include<stack>
8 #include<queue>
9 #include<math.h>
10 using namespace std;
11 #define int long long
12 #define INF 1000000000
13 #define LLINF 9223372036854775807
14 #define mod 1000000007
```

ソースコード 16　alias

```
1 typedef vector<int> VI;
2 typedef pair<int, int> pii;
3 typedef priority_queue<int> PQ;
4 #define SORT(c) sort((c).begin(),(c).end())
5 #define rSORT(c) sort((c).rbegin(),(c).rend())
```

ソースコード 17　ループ

```
#define fore(i,a) for(auto &i:a)
#define REP(i,n) for(int i=0;i<n;i++)
#define eREP(i,n) for(int i=0;i<=n;i++)
#define FOR(i,a,b) for(int i=(a);i<(b);++i)
#define eFOR(i,a,b) for(int i=(a);i<=(b);++i)
```

データ構造

ソースコード 18　UnionFind

```
class UnionFind {
  public:
    vector<int> Parent;
    UnionFind(int N) {
      Parent = vector<int>(N, -1);
    }
    int root(int A) {
      if (Parent[A] < 0) return A;
      return Parent[A] = root(Parent[A]);
    }
    int size(int A) {return -Parent[root(A)];}

    bool connect(int A, int B) {
      A = root(A);
      B = root(B);
      if (A == B) return false;

      if (size(A) < size(B)) swap(A, B);

      Parent[A] += Parent[B];
      Parent[B] = A;

      return true;
    }
};
```

ソースコード 19　セグ木

```
struct segtree {
  public:
    const int SIZE = 1 << 18;
    VI seg, lazy;
    misegtree() :seg(SIZE * 2, INF), lazy(SIZE * 2, INF) {}
    void lazy_evaluate(int k, int l, int r) {
      if (lazy[k] != 0) {
```

```
8         //seg[k] += lazy[k];
9         seg[k] = min(seg[k], lazy[k]);
10        if (r - l > 1) {
11          //lazy[k * 2 + 1] += lazy[k];
12          //lazy[k * 2 + 2] += lazy[k];
13          //lazy[k * 2 + 1] = min(lazy[k * 2 + 1], lazy[k]);
14          //lazy[k * 2 + 2] = min(lazy[k * 2 + 2], lazy[k]);
15          //lazy[k * 2 + 1] = max(lazy[k * 2 + 1], lazy[k]);
16          //lazy[k * 2 + 2] = max(lazy[k * 2 + 2], lazy[k]);
17        }
18        lazy[k] = LLINF;
19      }
20    }
21    void update(int a, int b, int k, int l, int r, int x) {
22      lazy_evaluate(k, l, r);
23      if (r <= a || b <= l)return;
24      if (a <= l && r <= b) {
25        //lazy[k] += x;
26        //lazy[k] = min(lazy[k], x);
27        //lazy[k] = max(lazy[k], x);
28        lazy_evaluate(k, l, r);
29      } else {
30        update(a, b, k * 2 + 1, l, (l + r) / 2, x);
31        update(a, b, k * 2 + 2, (l + r) / 2, r, x);
32        //seg[k] = seg[k * 2 + 1] + seg[k * 2 + 2];
33        //seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]);
34        //seg[k] = max(seg[k * 2 + 1], seg[k * 2 + 2]);
35      }
36    }
37    int query(int a, int b, int k, int l, int r) {
38      lazy_evaluate(k, l, r);
39      if (r <= a || b <= l)return LLINF;
40      if (a <= l && r <= b)return seg[k];
41      int x = query(a, b, k * 2 + 1, l, (l + r) / 2);
42      int y = query(a, b, k * 2 + 2, (l + r) / 2, r);
43      //return x + y;
44      //return min(x, y);
45      //return max(x, y);
46    }
47    void update(int a, int b, int x) { update(a, b, 0, 0, SIZE, x); }
48    int query(int a, int b) {
49      return query(a, b, 0, 0, SIZE);
50    }
51  };
```

グラフ

ソースコード 20　ダイクストラ

```
const int MAX_N = 100;
int G[100][100];
int N;
VI dist(110);

void dikstra(int s) {
  priority_queue<pii, vector<pii>, greater<pii> >que;
  dist[s] = 0;
  que.push(pii(0, s));
  while (!que.empty()) {
    pii p = que.top();
    que.pop();
    int v = p.second;
    if (dist[v] < p.first) continue;
    REP(i, N) {
      if (G[v][i] == LLINF) continue;
      if (dist[i] > dist[v] + G[v][i]) {
        dist[i] = dist[v] + G[v][i];
        que.push(pii(dist[v], i));
      }
    }
  }
}
```

ソースコード 21　強結合なんとか

```
const int MAX_N=10000;
vector<VI> g1(MAX_N);
vector<VI> g2(MAX_N);
VI flag(MAX_N);
int fcnt = 0;
int gcnt = 0;
vector<bool>vis(MAX_N, false);
VI group(MAX_N);

void dfs(int v) {
  vis[v] = true;
  for (int w : g1[v]) {
    if (!vis[w])dfs(w);
  }
  flag[fcnt] = v;
  //cout << fcnt << " " << v << endl;
```

```
  fcnt++;
}

void rdfs(int v) {
  vis[v] = false;
  group[v] = gcnt;
  //cout << v << " " << gcnt << endl;
  for (int w : g2[v]) {
    if (vis[w])rdfs(w);
  }
}
```

bound

ソースコード 22　bound

```
#define LB(x,a) lower_bound((x).begin(),(x).end(),(a))
#define UB(x,a) upper_bound((x).begin(),(x).end(),(a))
```

MOD

ソースコード 23　MOD 逆元

```
int modpow(int a, int p) {
  if (p == 0) return 1;
  if (p % 2 == 0) {
    int halfP = p / 2;
    int half = modpow(a, halfP);

    return half * half % mod;
  } else {
    return a * modpow(a, p - 1) % mod;
  }
}

int comb(int a, int b) {
  if (b > a - b) return comb(a, a - b);
  int ansMul = 1;
  int ansDiv = 1;
  for (int i = 0; i < b; i++) {
    ansMul *= (a - i);
    ansDiv *= (i + 1);
    ansMul %= mod;
    ansDiv %= mod;
  }
```

```
23
24    int ans = ansMul * modpow(ansDiv, mod - 2) % mod;
25    return ans;
26  }
```