Standard Code Library

Your TeamName

Your School

August 2, 2025

Contents

一切的开始

宏定义

- 需要 C++11

```cpp
#include <bits/stdc++.h>
using namespace std;
using LL = long long;
#define FOR(i, x, y) for (decay<decltype(y)>::type i = (x), _##i = (y); i < _##i; ++i)
#define FORD(i, x, y) for (decay<decltype(x)>::type i = (x), _##i = (y); i > _##i; --i)
#ifdef DEBUG
#ifndef ONLINE_JUDGE
#define zerol
#endif
#endif
#ifdef zerol
#define dbg(x...) do { cout << "\033[32;1m" << #x << " -> "; err(x); } while (0)
void err() { cout << "\033[39;0m" << endl; }
template<template<typename...> class T, typename t, typename... A>
void err(T<t> a, A... x) { for (auto v: a) cout << v << ' '; err(x...); }
template<typename T, typename... A>
void err(T a, A... x) { cout << a << ' '; err(x...); }
#else
#define dbg(...)
#define err(...)
#endif
// ------------------------------------------------------------------
```

- 调试时添加编译选项 -DDEBUG，提交时注释
- 注意检查判题系统编译选项，修改 #ifndef ONLINE_JUDGE
- FOR ++ 循环 FOR(循环变量名称，循环变量起始值，循环变量结束值（不含）)
- FORD -循环
- err() 调试时输出（支持单层迭代）
- dbg() 变色输出变量名和变量值（支持单层迭代）
- 黄色 33, 蓝色 34, 橙色 31

对拍

- Linux

```bash
#!/usr/bin/env bash
g++ -o r main.cpp -O2 -std=c++11
g++ -o std std.cpp -O2 -std=c++11
while true; do
    python gen.py > in
    ./std < in > stdout
    ./r < in > out
    if test $? -ne 0; then
        exit 0
    fi
    if diff stdout out; then
        printf "AC\n"
    else
        printf "GG\n"
        exit 0
    fi
done
```

- Windows

```
@echo off
setlocal enabledelayedexpansion

g++ -o r main.cpp -O2 -std=c++11
g++ -o std std.cpp -O2 -std=c++11

:loop
python gen.py > in
if !errorlevel! neq 0 exit /b
```

```bat
10
11   std.exe < in > stdout
12   if !errorlevel! neq 0 exit /b
13
14   r.exe < in > out
15   if !errorlevel! neq 0 exit /b
16
17   fc /b stdout out > nul
18   if !errorlevel! equ 0 (
19       echo AC
20   ) else (
21       echo GG
22       exit /b
23   )
24
25   goto loop
```

快速编译运行（配合无插件 VSC）

- Linux

```bash
1  #!/bin/bash
2  g++ $1.cpp -o $1 -O2 -std=c++14 -Wall -Dzerol -g
3  if $? -eq 0; then
4      ./$1
5  fi
```

- Windows

```bat
@echo off
:: 参数为文件名（不含.cpp后缀）
g++ %1.cpp -o %1 -O2 -std=c++14 -Wall -Dzerol -g
if %errorlevel% equ 0 (
    %1.exe
)
```

数据结构

ST 表

- 一维

```cpp
1  #define M 10
2
3  struct RMQ {
4      int f[22][M];
5      inline int highbit(int x) { return 31 - __builtin_clz(x); }
6      void init(int* v, int n) {
7          FOR (i, 0, n) f[0][i] = v[i];
8          FOR (x, 1, highbit(n) + 1)
9          FOR (i, 0, n - (1 << x) + 1)
10         f[x][i] = min(f[x - 1][i], f[x - 1][i + (1 << (x - 1))]);
11     }
12     int get_min(int l, int r) {
13         assert(l <= r);
14         int t = highbit(r - l + 1);
15         return min(f[t][l], f[t][r - (1 << t) + 1]);
16     }
17 };
```

- 二维

```cpp
1  #define maxn 10
2  LL n, m, a[maxn][maxn];
3
4  struct RMQ2D{
5      int f[maxn][maxn][10][10];
6      inline int highbit(int x) { return 31 - __builtin_clz(x); }
```

```
7        inline int calc(int x, int y, int xx, int yy, int p, int q) {
8            return max(
9                max(f[x][y][p][q], f[xx - (1 << p) + 1][yy - (1 << q) + 1][p][q]),
10               max(f[xx - (1 << p) + 1][y][p][q], f[x][yy - (1 << q) + 1][p][q])
11               );
12       }
13       void init() {
14           FOR (x, 0, highbit(n) + 1)
15           FOR (y, 0, highbit(m) + 1)
16           FOR (i, 0, n - (1 << x) + 1)
17           FOR (j, 0, m - (1 << y) + 1) {
18               if (!x && !y) { f[i][j][x][y] = a[i][j]; continue; }
19               f[i][j][x][y] = calc(
20                   i, j,
21                   i + (1 << x) - 1, j + (1 << y) - 1,
22                   max(x - 1, 0), max(y - 1, 0)
23                   );
24           }
25       }
26       inline int get_max(int x, int y, int xx, int yy) {
27           return calc(x, y, xx, yy, highbit(xx - x + 1), highbit(yy - y + 1));
28       }
29   };
```

线段树

朴素线段树

- 默认为最大值, 可自行修改 struct Q struct P P operator &
- 注意建树时的下标问题 (1-based)

```
1    const LL INF = LONG_LONG_MAX;
2    #define maxn 10
3    LL n;
4
5    namespace SGT {
6        struct Q {
7            LL setv;
8            explicit Q(LL setv = -1): setv(setv) {}
9            void operator += (const Q& q) { if (q.setv != -1) setv = q.setv; }
10       };
11       struct P {
12           LL max;
13           explicit P(LL max = -INF): max(max) {}
14           void up(Q& q) { if (q.setv != -1) max = q.setv; }
15       };
16       template<typename T>
17       P operator & (T&& a, T&& b) {
18           return P(max(a.max, b.max));
19       }
20       P p[maxn << 2];
21       Q q[maxn << 2];
22   #define lson o * 2, l, (l + r) / 2
23   #define rson o * 2 + 1, (l + r) / 2 + 1, r
24       void up(int o, int l, int r) {
25           if (l == r) p[o] = P();
26           else p[o] = p[o * 2] & p[o * 2 + 1];
27           p[o].up(q[o]);
28       }
29       void down(int o, int l, int r) {
30           q[o * 2] += q[o]; q[o * 2 + 1] += q[o];
31           q[o] = Q();
32           up(lson); up(rson);
33       }
34       template<typename T>
35       void build(T&& f, int o = 1, int l = 1, int r = n) {
36           if (l == r) q[o] = f(l);
37           else { build(f, lson); build(f, rson); q[o] = Q(); }
38           up(o, l, r);
39       }
40       P query(int ql, int qr, int o = 1, int l = 1, int r = n) {
```

```cpp
            if (ql > r || l > qr) return P();
            if (ql <= l && r <= qr) return p[o];
            down(o, l, r);
            return query(ql, qr, lson) & query(ql, qr, rson);
        }
        void update(int ql, int qr, const Q& v, int o = 1, int l = 1, int r = n) {
            if (ql > r || l > qr) return;
            if (ql <= l && r <= qr) q[o] += v;
            else {
                down(o, l, r);
                update(ql, qr, v, lson); update(ql, qr, v, rson);
            }
            up(o, l, r);
        }
    }

    // -------------------------------------------------------------------
    void solve(){
        vector<LL> arr = {1, 5, 7, 4, 2, 8 ,3, 6, 10, 9};
        n = arr.size();
        SGT::build([&](int idx){
            return SGT::Q(arr[idx-1]);
        });
        for(LL i=1; i<=n; i++){
            dbg(SGT::query(1, i).max);
        }
        SGT::update(2, 4, SGT::Q(-3));
        cout << "MODIFIED\n";
        for(LL i=1; i<=n; i++){
            dbg(SGT::query(1, i).max);
        }
    }
```

  ● 区间修改，区间累加，查询区间和、最大值、最小值。

```cpp
#define maxn 100005
#define INF LONG_LONG_MAX
LL a[maxn], n;

struct IntervalTree {
#define ls o * 2, l, m
#define rs o * 2 + 1, m + 1, r
    static const LL M = maxn * 4, RS = 1E18 - 1;
    LL addv[M], setv[M], minv[M], maxv[M], sumv[M];
    void init() {
        memset(addv, 0, sizeof addv);
        fill(setv, setv + M, RS);
        memset(minv, 0, sizeof minv);
        memset(maxv, 0, sizeof maxv);
        memset(sumv, 0, sizeof sumv);
    }
    void maintain(LL o, LL l, LL r) {
        if (l < r) {
            LL lc = o * 2, rc = o * 2 + 1;
            sumv[o] = sumv[lc] + sumv[rc];
            minv[o] = min(minv[lc], minv[rc]);
            maxv[o] = max(maxv[lc], maxv[rc]);
        } else sumv[o] = minv[o] = maxv[o] = 0;
        if (setv[o] != RS) { minv[o] = maxv[o] = setv[o]; sumv[o] = setv[o] * (r - l + 1); }
        if (addv[o]) { minv[o] += addv[o]; maxv[o] += addv[o]; sumv[o] += addv[o] * (r - l + 1); }
    }
    void build(LL o, LL l, LL r) {
        if (l == r) addv[o] = a[l];
        else {
            LL m = (l + r) / 2;
            build(ls); build(rs);
        }
        maintain(o, l, r);
    }
    void pushdown(LL o) {
        LL lc = o * 2, rc = o * 2 + 1;
        if (setv[o] != RS) {
```

```
38              setv[lc] = setv[rc] = setv[o];
39              addv[lc] = addv[rc] = 0;
40              setv[o] = RS;
41          }
42          if (addv[o]) {
43              addv[lc] += addv[o]; addv[rc] += addv[o];
44              addv[o] = 0;
45          }
46      }
47      void update(LL p, LL q, LL o, LL l, LL r, LL v, LL op) {
48          if (p <= r && l <= q){
49              if (p <= l && r <= q) {
50                  if (op == 2) { setv[o] = v; addv[o] = 0; }
51                  else addv[o] += v;
52              } else {
53                  pushdown(o);
54                  LL m = (l + r) / 2;
55                  update(p, q, ls, v, op); update(p, q, rs, v, op);
56              }
57          }
58          maintain(o, l, r);
59      }
60      void query(LL p, LL q, LL o, LL l, LL r, LL add, LL& ssum, LL& smin, LL& smax) {
61          if (p > r || l > q) return;
62          if (setv[o] != RS) {
63              LL v = setv[o] + add + addv[o];
64              ssum += v * (min(r, q) - max(l, p) + 1);
65              smin = min(smin, v);
66              smax = max(smax, v);
67          } else if (p <= l && r <= q) {
68              ssum += sumv[o] + add * (r - l + 1);
69              smin = min(smin, minv[o] + add);
70              smax = max(smax, maxv[o] + add);
71          } else {
72              LL m = (l + r) / 2;
73              query(p, q, ls, add + addv[o], ssum, smin, smax);
74              query(p, q, rs, add + addv[o], ssum, smin, smax);
75          }
76      }
77      // 简化接口
78      void build(int n) {
79          build(1, 1, n);
80      }
81
82      void range_add(int l, int r, int val) {
83          update(l, r, 1, 1, n, val, 1);
84      }
85
86      void range_set(int l, int r, int val) {
87          update(l, r, 1, 1, n, val, 2);
88      }
89
90      void range_query(int l, int r, LL& sum, LL& min_val, LL& max_val) {
91          sum = 0;
92          min_val = INF;
93          max_val = -INF;
94          query(l, r, 1, 1, n, 0, sum, min_val, max_val);
95      }
96  } IT;
97  // ----------------------------------------------------------------
98  void solve(){
99      IT.init();
100
101     n = 5;
102     vector<int> data = {1, 3, 5, 7, 9};
103     for (int i = 0; i < n; i++) {
104         a[i + 1] = data[i];   // 注意: 线段树从 1 开始索引
105     }
106
107     IT.build(n);
108
```

```
109      LL sum, min_val, max_val;
110      IT.range_query(1, 5, sum, min_val, max_val);
111      cout << " " << sum << " " << min_val << " " << max_val << endl;
112
113      IT.range_add(2, 4, 2);
114      IT.range_query(1, 5, sum, min_val, max_val);
115      cout << " " << sum << " " << min_val << " " << max_val << endl;
116
117      IT.range_set(3, 5, 10);
118      IT.range_query(1, 5, sum, min_val, max_val);
119      cout << " " << sum << " " << min_val << " " << max_val << endl;
120
121      IT.range_query(2, 4, sum, min_val, max_val);
122      cout << " " << sum << " " << min_val << " " << max_val << endl;
123  }
```

数学

图论

计算几何

字符串

杂项