```
线段树
单点修改 区间查询

#define maxn 50005
struct node
{
    int l, r;
    int sum;
}a[maxn << 4];

void build(int l, int r, int rt)
{
    int m = (l + r) >> 1;
    a[rt].l = l;
    a[rt].r = r;
    if (l == r)
    {
        cin >> a[rt].sum;
        return;
    }
    build(l, m, rt << 1);
    build(m + 1, r, rt << 1 | 1);
    a[rt].sum = a[rt << 1].sum + a[rt << 1 | 1].sum;
}
void update(int p, int add, int l, int r, int rt)
{
    if (l == r)
    {
        a[rt].sum += add;
        return;
    }
    int m = (l + r) >> 1;
    if (p <= m)
        update(p, add, l, m, rt << 1);
    else
        update(p, add, m + 1, r, rt << 1 | 1);
    a[rt].sum = a[rt << 1].sum + a[rt << 1 | 1].sum;
}

int query(int l, int r, int rt)
{
    if (l <= a[rt].l&&a[rt].r <= r)
        return a[rt].sum;
    int m = (a[rt].l + a[rt].r) >> 1, ans = 0;
    if (l <= m)
        ans += query(l, r, rt << 1);
    if (r>m)
        ans += query(l, r, rt << 1 | 1);
    return ans;
}
int main()
{
    IOS;
    int n;
    int _;
    cin >> _;
```

```
57      for (int cas = 1; cas <= _; cas++)
58      {
59          cout << "Case " << cas << ":\n";
60          cin >> n;
61          build(1, n, 1);
62          string str;
63          while (cin >> str)
64          {
65              int a, b;
66              if (str[0] == 'E')
67                  break;
68              cin >> a >> b;
69              if (str[0] == 'Q')
70                  cout << query(a, b, 1) << endl;
71              else
72              {
73                  if (str[0] == 'S')
74                      b = -b;
75                  update(a, b, 1, n, 1);
76              }
77          }
78      }
79  }
80
81  区间修改 区间查询
82
83
84  #define maxn 100007
85  struct node
86  {
87      int l, r, m;
88      ll sum, lazy;
89  } a[maxn << 2];
90  void build(int l, int r, int rt)
91  {
92      a[rt].l = l;
93      a[rt].r = r;
94      a[rt].m = (l + r) >> 1;
95      a[rt].lazy = 0;
96      if (l == r)
97      {
98          cin >> a[rt].sum;
99          return;
100     }
101     int m = (l + r) >> 1;
102     build(l, m, lson);
103     build(m + 1, r, rson);
104     a[rt].sum = (a[lson].sum + a[rson].sum);
105 }
106 void update(int l, int r, int val, int rt)
107 {
108     if (a[rt].l == l && a[rt].r == r)
109     {
110         a[rt].lazy += val;
111         return;
112     }
```

```cpp
    a[rt].sum += val * (r - l + 1);
    if (a[rt].m >= r)
        update(l, r, val, lson);
    else if (a[rt].m<l)
        update(l, r, val, rson);
    else
    {
        update(l, a[rt].m, val, lson);
        update(a[rt].m + 1, r, val, rson);
    }
}
void pushdown(int rt)
{
    if (a[rt].lazy)
    {
        a[lson].lazy += a[rt].lazy;
        a[rson].lazy += a[rt].lazy;
        a[rt].sum += (a[rt].r - a[rt].l + 1)*a[rt].lazy;
        a[rt].lazy = 0;
    }
}
ll query(int l, int r, int rt)
{
    if (a[rt].l == l && a[rt].r == r)
        return a[rt].sum + a[rt].lazy*(r - l + 1);
    pushdown(rt);
    if (a[rt].m >= r)
        return query(l, r, lson);
    else if (a[rt].m<l)
        return query(l, r, rson);
    else
        return query(l, a[rt].m, lson) + query(a[rt].m + 1, r, rson);
}
int main()
{
    IOS;
    int n, q;
    while (cin >> n >> q)
    {
        build(1, n, 1);
        string str;
        for (int i = 0; i<q; i++)
        {
            cin >> str;
            int l, r, v;
            if (str[0] == 'Q')
            {
                cin >> l >> r;
                cout << query(l, r, 1) << Endl;
            }
            else if (str[0] == 'C')
            {
                cin >> l >> r >> v;
                update(l, r, v, 1);
            }
        }
```

```
169     }
170 }
171
172 区间修改为函数值 区间查询
173
174 const int maxn = 1000007;
175 struct t
176 {
177     int l, r, m;
178     ll v;
179 }a[maxn << 2];
180 void build(int l, int r, int rt)
181 {
182     a[rt].l = l;
183     a[rt].r = r;
184     int m = (l + r) >> 1;
185     a[rt].m = m;
186     if (l == r)
187     {
188         cin >> a[rt].v;
189         return;
190     }
191     build(l, m, rt << 1);
192     build(m + 1, r, rt << 1 | 1);
193     a[rt].v = a[rt << 1].v + a[rt << 1 | 1].v;
194 }
195 ll query(ll l, ll r, ll rt)
196 {
197     if (a[rt].l >= l && a[rt].r <= r)
198         return a[rt].v;
199     ll sum = 0;
200     if (l <= a[rt].m)
201         sum += query(l, r, rt << 1);
202     if (r > a[rt].m)
203         sum += query(l, r, rt << 1 | 1);
204     return sum;
205 }
206 void update(int l, int r, int rt)
207 {
208     if (a[rt].l == a[rt].r)
209     {
210         a[rt].v = (ll)sqrt(a[rt].v);
211         return;
212     }
213     if (a[rt].l >= l && a[rt].r <= r && a[rt].v == a[rt].l - a[rt].r + 1)
214         return;
215     if (l <= a[rt].m)
216         update(l, r, rt << 1);
217     if (r > a[rt].m)
218         update(l, r, rt << 1 | 1);
219     a[rt].v = a[rt << 1].v + a[rt << 1 | 1].v;
220 }
221 int main()
222 {
223     IOS;
224     int t = 1, n;
```

```
225     while (cin >> n)
226     {
227         build(1, n, 1);
228         int q;
229         cin >> q;
230         cout << "Case #" << t++ << ":\n";
231         while (q--)
232         {
233             int o, l, r;
234             cin >> o >> l >> r;
235             if (l>r)
236                 swap(l, r);
237             if (o)
238                 cout << query(l, r, 1) << Endl;
239             else update(l, r, 1);
240         }
241         cout << '\n';
242     }
243 }
244
245 区间线段树 单点破坏 / 修复 查询单点所在区间长度
246
247 const int maxn = 50000 + 10;
248
249 int n, m;
250 int s[maxn], top;//s为模拟栈
251
252 struct node
253 {
254     int l, r;
255     int ls, rs, ms;//ls,左端最大连续区间，rs右端最大连续区间，ms区间内最大连续区间
256 } a[maxn << 2];
257
258 void build(int l, int r, int rt)
259 {
260     a[rt].l = l;
261     a[rt].r = r;
262     a[rt].ls = a[rt].rs = a[rt].ms = r - l + 1;
263     if (l != r)
264     {
265         int mid = (l + r) >> 1;
266         build(l, mid, lson);
267         build(mid + 1, r, rson);
268     }
269 }
270
271 void update(int p, int v, int rt)
272 {
273     if (a[rt].l == a[rt].r)
274     {
275         if (v == 1)
276             a[rt].ls = a[rt].rs = a[rt].ms = 1;
277         else
278             a[rt].ls = a[rt].rs = a[rt].ms = 0;
279         return;
280     }
```

```
281     int mid = (a[rt].l + a[rt].r) >> 1;
282     if (p <= mid)
283         update(p, v, lson);
284     else
285         update(p, v, rson);
286     a[rt].ls = a[lson].ls;
287     a[rt].rs = a[rson].rs;
288     a[rt].ms = max(max(a[lson].ms, a[rson].ms), a[lson].rs + a[rson].ls);
289     if (a[lson].ls == a[lson].r - a[lson].l + 1)
290         a[rt].ls += a[rson].ls;
291     if (a[rson].rs == a[rson].r - a[rson].l + 1)
292         a[rt].rs += a[lson].rs;
293 }
294
295 int query(int p, int rt)
296 {
297     if (a[rt].l == a[rt].r || a[rt].ms == 0 || a[rt].ms == a[rt].r - a[rt].l + 1)
298         return a[rt].ms;
299     int mid = (a[rt].l + a[rt].r) >> 1;
300     if (p <= mid)
301     {
302         if (p >= a[lson].r - a[lson].rs + 1)
303             return query(p, lson) + query(mid + 1, rson);
304         else
305             return query(p, lson);
306     }
307     else
308     {
309         if (p <= a[rson].l + a[rson].ls - 1)
310             return query(p, rson) + query(mid, lson);
311         else
312             return query(p, rson);
313     }
314 }
315
316 int main()
317 {
318     int i, j, x;
319     char ch[2];
320     while (~scanf("%d%d", &n, &m))
321     {
322         top = 0;
323         build(1, n, 1);
324         while (m--)
325         {
326             scanf("%s", ch);
327             if (ch[0] == 'D')
328             {
329                 scanf("%d", &x);
330                 s[top++] = x;
331                 update(x, 0, 1);
332             }
333             else if (ch[0] == 'Q')
334             {
335                 scanf("%d", &x);
336                 printf("%d\n", query(x, 1));
```

```
337             }
338             else
339             {
340                 if (x>0)
341                 {
342                     x = s[--top];
343                     update(x, 1, 1);
344                 }
345             }
346         }
347     }
348 }
349
350 约会安排 带优先级 （两棵树）
351 #include<bits/stdc++.h>
352 using namespace std;
353 #define mem(a,b) memset((a),b,sizeof((a)))
354 #define clr(sum) (sum).clear()
355 #define mp make_pair
356 #define pb push_back
357 #define ll long long
358 #define ld long double
359 #define Endl '\n'
360 #define IOS ios::sync_with_stdio(0);cin.tie(0);cout.tie(0)
361 #define lowbit(i) (i&(-i))
362 #define lson rt<<1
363 #define rson lson|1
364
365
366 #define maxn 50005
367
368 struct
369 {
370     int l, r, m;
371     int lds;
372     int rds;
373     int mds;
374     int lns;
375     int rns;
376     int mns;
377     int coverds;
378     int coverns;
379 }a[maxn << 2];
380 void build(int l, int r, int rt)
381 {
382     a[rt].lds = a[rt].rns = a[rt].rds = a[rt].lns = a[rt].mds = a[rt].mns = r - l + 1;
383     a[rt].coverds = a[rt].coverns = -1;
384     a[rt].l = l;
385     a[rt].r = r;
386     int m = (l + r) >> 1;
387     a[rt].m = m;
388     if (l == r)
389         return;
390     build(l, m, lson);
391     build(m + 1, r, rson);
392 }
```

```
393 void push_up(int len, int p, int rt)
394 {
395     if (p == 0)
396     {
397         a[rt].lds = a[lson].lds;
398         a[rt].rds = a[rson].rds;
399         if (a[rt].lds == len - (len >> 1))
400             a[rt].lds += a[rson].lds;
401         if (a[rt].rds == (len >> 1))
402             a[rt].rds += a[lson].rds;
403         a[rt].mds = max(a[lson].rds + a[rson].lds, max(a[lson].mds, a[rson].mds));
404     }
405     else
406     {
407         a[rt].lds = a[lson].lds;
408         a[rt].rds = a[rson].rds;
409         if (a[rt].lds == len - (len >> 1))
410             a[rt].lds += a[rson].lds;
411         if (a[rt].rds == (len >> 1))
412             a[rt].rds += a[lson].rds;
413         a[rt].mds = max(a[lson].rds + a[rson].lds, max(a[lson].mds, a[rson].mds));
414         a[rt].lns = a[lson].lns;
415         a[rt].rns = a[rson].rns;
416         if (a[rt].lns == len - (len >> 1))
417             a[rt].lns += a[rson].lns;
418         if (a[rt].rns == (len >> 1))
419             a[rt].rns += a[lson].rns;
420         a[rt].mns = max(a[lson].rns + a[rson].lns, max(a[lson].mns, a[rson].mns));
421     }
422 }
423 void push_down(int rt, int len)
424 {
425     if (a[rt].coverds != -1)
426     {
427         a[lson].coverds = a[rson].coverds = a[rt].coverds;
428         a[lson].lds = a[lson].rds = a[lson].mds = a[rt].coverds ? 0 : len - (len >>
              1);
429         a[rson].lds = a[rson].rds = a[rson].mds = a[rt].coverds ? 0 : (len >> 1);
430         a[rt].coverds = -1;
431     }
432     if (a[rt].coverns != -1)
433     {
434         a[lson].coverns = a[rson].coverns = a[rt].coverns;
435         a[lson].lns = a[lson].rns = a[lson].mns = a[rt].coverns ? 0 : len - (len >>
              1);
436         a[rson].lns = a[rson].rns = a[rson].mns = a[rt].coverns ? 0 : (len >> 1);
437         a[rt].coverns = -1;
438     }
439 }
440 int query(int pos, int p, int rt)
441 {
442     int l = a[rt].l, r = a[rt].r;
443     if (l == r)
444         return l;
445     push_down(rt, r - l + 1);
446     int m = (l + r) >> 1;
```

```
447     if (p == 0)
448     {
449         if (pos <= a[lson].mds)
450             return query(pos, p, lson);
451         else if (a[lson].rds + a[rson].lds >= pos)
452             return m - a[lson].rds + 1;
453         else
454             return query(pos, p, rson);
455     }
456     else
457     {
458         if (pos <= a[lson].mns)
459             return query(pos, p, lson);
460         else if (a[lson].rns + a[rson].lns >= pos)
461             return m - a[lson].rns + 1;
462         else
463             return query(pos, p, rson);
464     }
465 }
466 void update(int L, int R, int p, int c, int rt)
467 {
468     int l = a[rt].l, r = a[rt].r;
469     if (L <= l && R >= r)
470     {
471         if (p == 0)
472         {
473             a[rt].lds = a[rt].rds = a[rt].mds = c ? 0 : r - l + 1;
474             a[rt].coverds = c;
475         }
476         else
477         {
478             a[rt].lns = a[rt].rns = a[rt].mns = c ? 0 : r - l + 1;
479             a[rt].coverns = c;
480         }
481         return;
482     }
483     push_down(rt, r - l + 1);
484     int m = (l + r) >> 1;
485     if (L <= m)
486         update(L, R, p, c, lson);
487     if (R>m)
488         update(L, R, p, c, rson);
489     push_up(r - l + 1, p, rt);
490 }
491 int main()
492 {
493     int i, t, m, n, time, L, R, ans;
494     char op[10];
495     scanf("%d", &t);
496     for (i = 1; i <= t; i++)
497     {
498         L = R = 0;
499         scanf("%d%d", &n, &m);
500         build(1, n, 1);
501         printf("Case %d:\n", i);
502         while (m--)
```

```
503         {
504             scanf("%s", op);
505             switch (op[0])
506             {
507             case 'D':
508                 scanf("%d", &time);
509                 if (time>a[1].mds)
510                     printf("fly with yourself\n");
511                 else
512                 {
513                     ans = query(time, 0, 1);
514                     printf("%d,let's fly\n", ans);
515                     update(ans, ans + time - 1, 0, 1, 1);
516                 }
517                 break;
518             case 'N':
519                 scanf("%d", &time);
520                 if (time > a[1].mds)
521                 {
522                     if (time > a[1].mns)
523                         printf("wait for me\n");
524                     else
525                     {
526                         ans = query(time, 1, 1);
527                         printf("%d,don't put my gezi\n", ans);
528                         update(ans, ans + time - 1, 0, 1, 1);
529                         update(ans, ans + time - 1, 1, 1, 1);
530                     }
531                 }
532                 else
533                 {
534                     ans = query(time, 0, 1);
535                     printf("%d,don't put my gezi\n", ans);
536                     update(ans, ans + time - 1, 0, 1, 1);
537                     update(ans, ans + time - 1, 1, 1, 1);
538                 }
539                 break;
540             case 'S':
541                 scanf("%d%d", &L, &R);
542                 printf("I am the hope of chinese chengxuyuan!!\n");
543                 update(L, R, 0, 0, 1);
544                 update(L, R, 1, 0, 1);
545                 break;
546             }
547         }
548     }
549     return 0;
550 }
551
```