线段树维护最大子段和

主讲人：邓哲也

SPOJ GSS1 Can you answer these queries I

你有一个长度为 n 的序列 A[1], A[2], …, A[N].

询问:

Query(x, y) = max { A[i] + … + A[j]; x <= i <= j <= y}

给出 M 组 (x, y), 请给出 M 次询问的答案。

|A[i]| <= 15007, 1 <= N,M <= 50000

SPOJ GSS1 Can you answer these queries I

考虑用线段树来维护每个区间的答案。

假设 smax[x] 表示区间 x 的答案。

那么 smax[x] 如何由 smax[ls] 和 smax[rs] 合并得来呢?

不能直接合并。

我们还需要记录每个区间的前缀和最大值 lmax[x] 和后缀和最大值 rmax[x]。

此时: smax[x] = max(max(smax[ls], smax[rs]), rmax[ls] + lmax[rs])

SPOJ GSS1 Can you answer these queries I

为了更新 lmax 和 rmax，我们还需要记录每个区间的区间和 sum。

这样就有：

lmax[x] = max(lmax[ls], sum[ls] + lmax[rs])

rmax[x] = max(rmax[rs], sum[rs] + rmax[ls])

SPOJ GSS1 Can you answer these queries I

```
void update(int x){
    smax[x] = max(max(smax[ls], smax[rs]), rmax[ls] +
lmax[rs]);
    lmax[x] = max(lmax[ls], sum[ls] + lmax[rs]);
    rmax[x] = max(rmax[rs], sum[rs] + rmax[ls]);
    sum[x] = sum[ls] + sum[rs];
}
```

SPOJ GSS1 Can you answer these queries I

```
void build(int l, int r, int x){
    if (l == r){
        smax[x] = lmax[x] = rmax[x] = sum[x] = a[l];
        return;
    }
    int mid = (l + r) >> 1;
    build(l, mid, ls);
    build(mid + 1, r, rs);
    update(x);
}
```

SPOJ GSS1 Can you answer these queries I

```
void query(int A, int B, int l, int r, int x, int
&Smax,
           int &Lmax, int &Rmax, int &Sum){
    if (A <= l && r <= B){
        Smax = smax[x];
        Lmax = lmax[x];
        Rmax = rmax[x];
        Sum = sum[x];
        return;
    }
```

SPOJ GSS1 Can you answer these queries I

```
void query(int A, int B, int l, int r, int x, int &Smax,
           int &Lmax, int &Rmax, int &Sum){
    if (A <= l && r <= B){
        …
    }
    int mid = (l + r) >> 1;
    int smaxl = -1e9, lmaxl = -1e9, rmaxl = -1e9, suml = 0;
    int smaxr = -1e9, lmaxr = -1e9, rmaxr = -1e9, sumr = 0;
    if (A <= mid) query(A, B, l, mid, ls, smaxl, lmaxl, rmaxl, suml);
    if (mid < B)  query(A, B, mid + 1, r, rs, smaxr, lmaxr, rmaxr,
sumr);
    Lmax = max(lmaxl, suml + lmaxr);
    Rmax = max(rmaxr, sumr + rmaxl);
    Smax = max(max(smaxl, smaxr), rmaxl + lmaxr);
    Sum = suml + sumr;
}
```

SPOJ GSS3 Can you answer these queries III

你有一个长度为 n 的序列 A[1], A[2], …, A[N].

0 x y: 把 A[x] 修改为 y (|y| <= 10000)

1 x y: Query(x, y) = max { A[i] + … + A[j]; x <= i <= j <= y}

给出 M 组操作，输出每次询问的答案

|A[i]| <= 10000, 1 <= N,M <= 50000

SPOJ GSS3 Can you answer these queries III

与 GSS1 不同的是，多了一个修改操作。

```
void modify(int p, int v, int l, int r, int x){
    if (l == r){
        smax[x] = lmax[x] = rmax[x] = sum[x] = v;
        return;
    }
    int mid = (l + r) >> 1;
    if (p <= mid) modify(p, v, l, mid, ls);
    else modify(p, v, mid + 1, r, rs);
    update(x);
}
```

SPOJ GSS5 Can you answer these queries V

你有一个长度为 n 的序列 A[1], A[2], …, A[N].

询问:

Query(x1, y1, x2, y2) = max { A[i] + … + A[j];

x1 <= i <= y1, x2 <= j <= y2}

x1 <= x2, y1 <= y2

给出 M 组操作，输出每次询问的答案

|A[i]| <= 10000, 1 <= N,M <= 10000

SPOJ GSS5 Can you answer these queries V

这次与以往不同的是，限定了左右端点的范围。

那么需要进行一下分类讨论。

如果 [x1, y1] 和 [x2, y2] 没有交集，即 $y1 < x2$

答案显然等于:

$$Rmax([x1, y1]) + Sum(y1 + 1, x2 - 1) + Lmax([x2, y2])$$

SPOJ GSS5 Can you answer these queries V

如果 [x1, y1] 和 [x2, y2] 有交集，即 y1 >= x2

这个时候，区间分为三个部分：

[x1, x2 - 1], [x2, y1], [y1 + 1 .. y2]

左端点有两种选择，右端点也有两种选择，一共四种情况。

进一步讨论，变为三种情况：

Smax([x2, y1])

Rmax([x1, x2 - 1]) + Lmax([x2, y2])

Rmax([x1, y1]) + Lmax([y1 + 1 .. y2])

下节课再见