

A题

恩，就是个线段树

建树的话 模板应该是

```
1 void build(int node, int l, int r)
2 {
3     int m = (l + r) / 2;
4     if (l == r)
5     {
6         segTree[node].first = data[l];
7         segTree[node].second = data[l];
8         return;
9     }
10    build(2 * node, l, m);
11    build(2 * node + 1, m + 1, r);
12    segTree[node] = calc(segTree[node * 2], segTree[node * 2 + 1]);
13 }
```

核心思路就是二分，直到 $l == r$ ，接着回溯时更新节点值

查询模板为

```
1 pair<int, int> query(int node, int l, int r, int ql, int qr)
2 {
3     pair<int, int> r1(0, 0), r2(0, 0);
4     int m = (l + r) / 2;
5     if (ql <= l && r <= qr)
6     {
7         return segTree[node];
8     }
9     if (ql <= m)
10    {
11        r1 = query(node * 2, l, m, ql, qr);
12    }
13    if (qr > m)
14    {
15        r2 = query(node * 2 + 1, m + 1, r, ql, qr);
16    }
17    return calc(r1, r2);
18 }
```

如果当前区间被查询区间包含，直接返回节点值

如果查询区间的左端在中点左边，查询左半部分

查询区间在中点右边，查询右半部分

以上

