

可持久化数组

```

struct PTA{
    vector<int> root;
    vector<int> lc, rc, tree;
    int cnt, rcnt, n;

private:
    int querynums(int l, int r, int p, int x){
        if(l == r) return tree[p];
        int mid = (l + r) >> 1;
        if(x <= mid) return querynums(l, mid, lc[p], x);
        else return querynums(mid + 1, r, rc[p], x);
    }
    int updatenums(int l, int r, int p, int x, int y){
        auto clone = [&](int p) -> int {
            cnt++;
            lc[cnt] = lc[p], rc[cnt] = rc[p];
            tree[cnt] = tree[p];
            return cnt;
        };
        p = clone(p);
        if(l == r){
            tree[p] = y;
            return p;
        }
        int mid = (l + r) >> 1;
        if(x <= mid) lc[p] = updatenums(l, mid, lc[p], x, y);
        else rc[p] = updatenums(mid + 1, r, rc[p], x, y);
        return p;
    }

public:
    PTA(int n1, int m1, vector<int> nums){
        root.resize(m1 + 1);
        lc.resize(n1 * 4 + m1 * 25);
        rc.resize(n1 * 4 + m1 * 25);
        tree.resize(n1 * 4 + m1 * 25);
        cnt = rcnt = 0; n = n1;
        auto build = [&](auto self, int l, int r, int p) -> int {
            p = ++cnt;
            if(l == r){
                tree[p] = nums[l];
                return p;
            }
            int mid = (l + r) >> 1;
            lc[p] = self(self, l, mid, lc[p]);
            rc[p] = self(self, mid + 1, r, rc[p]);
            return p;
        };
        root[0] = build(build, 1, n1, 0);
    }
    void update(int x, int y, int p){
        root[rcnt + 1] = updatenums(1, n, root[p], x, y);
        rcnt ++;
    }
    void update(int x, int y){
        update(x, y, rcnt);
    }
    void back(int x){
        root[rcnt + 1] = root[x];
        rcnt ++;
    }

```

```
    }  
    void back(){  
        back(rcnt);  
    }  
    int query(int x, int p){  
        return querynums(1, n, root[p], x);  
    }  
    int query(int x){  
        return query(x, rcnt);  
    }  
};
```

将可持久化数组简化成操作复杂度为 $O(\log n)$ 的数组，只用管接口就可以了。

初始化通过初始数据规模以及操作次数计算出 $vector$ 大小，更新询问时候如果不带有版本号默认最新版本。