线段树维护区间可合并信息

主讲人：邓哲也

POJ 2777 Count Color

对区间 [1, n] 进行两种操作：

1. C a b t： 给区间 [a, b] 染色为 t
2. P a b：查询区间[a, b]中有多少种不同的颜色

n, m <= 100000, 1 <= t <= 30

Smaple Input	Sample Output
2 2 4	2
C 1 1 2	1
P 1 2	
C 2 2 2	
P 1 2	

POJ 2777 Count Color

用线段树来维护一个区间中有几种颜色

这样是不够的。

因为光知道两个区间中各有几种颜色，是没法合并的。

POJ 2777 Count Color

用线段树来维护一个区间有哪几种颜色

考虑到颜色总数只有 30 个，可以用二进制位来表示颜色

每个区间维护一个 mask，若 mask 的第 i 位为 1，就表示这个区间里有第 i 种颜色。

查询的时候得到这个区间里颜色的二进制串，统计有几个一即可。

POJ 2777 Count Color

```
void update(int x){
	mask[x] = mask[ls] | mask[rs];
}
void build(int l, int r, int x){
	if (l == r){
		mask[x] = 1;
		return;
	}
	int mid = (l + r) >> 1;
	build(l, mid, ls);
	build(mid + 1, r, rs);
	update(x);
}
```

POJ 2777 Count Color

```
void down(int l, int r, int x){
        if(tag[x] != 0){
                tag[ls] = tag[rs] = tag[x];
                mask[ls] = mask[rs] = 1 << tag[x];
                tag[x] = 0;
        }
}
```

POJ 2777 Count Color

```
void change(int A, int B, int t, int l, int r, int x){
        if (A <= l && r <= B){
                mask[x] = 1 << t;
                tag[x] = t;
                return;
        }
        down(l, r, x);
        int mid = (l + r) >> 1;
        if (A <= mid) change(A, B, t, l, mid, ls);
        if (mid < B)  change(A, B, t, mid + 1, r, rs);
        update(x);
}
```

POJ 2777 Count Color

```
int query(int A, int B, int l, int r, int x){
      if (A <= l && r <= B) return mask[x];
      down(l, r, x);
      int mid = (l + r) >> 1, ret = 1;
      if (A <= mid) ret |= query(A, B, l, mid, ls);
      if (mid < B)  ret |= query(A, B, mid + 1, r, rs);
      return ret;
}

int ans = query(l, r, 1, n, 1);
```

真正的答案是 ans 的二进制表示中 1 的个数 ！

POJ 2777 Count Color

bitcount[x] 表示 x 的二进制表示中 1 的个数。

```
bitcount[1] = 1
for (int i = 2;i <= (1 << 16);i ++)
        bitcount[i] = bitcount[i >> 1] + (i & 1);
```

只需要计算:

```
bitcount[ans >> 16] + bitcount[ans & ((1 << 16) - 1)]
```

下节课再见