线段树介绍

主讲人：邓哲也

线段树的引入

在前面的章节中，我们学习了二叉排序树等一些树形结构。

基本思想都是对数据进行某种方式的划分。

树中的节点代表的一般都是单个元素。

线段树：

根据元素的下标进行划分

一个节点代表一个区间的信息

线段树的引入

问题：有一个长度为 n 的序列，a[1]，a[2]，…，a[n]。

现在执行 m 次操作，每次可以执行以下两种操作之一：

1. 将数列中的某个数 a[i] 修改为 v。
2. 询问一个下标区间 [l, r] 中所有数的和。

线段树的引入

朴素的做法：

直接用一个数组 a 维护。

修改：直接修改 a[i]，时间复杂度 O(1)

查询：枚举下标，时间复杂度 O(n)

线段树的引入

上述做法的瓶颈在于，我们维护的是以元素为单位的。

但是每次查询是一个区间。

所以我们也应该考虑直接去维护区间。

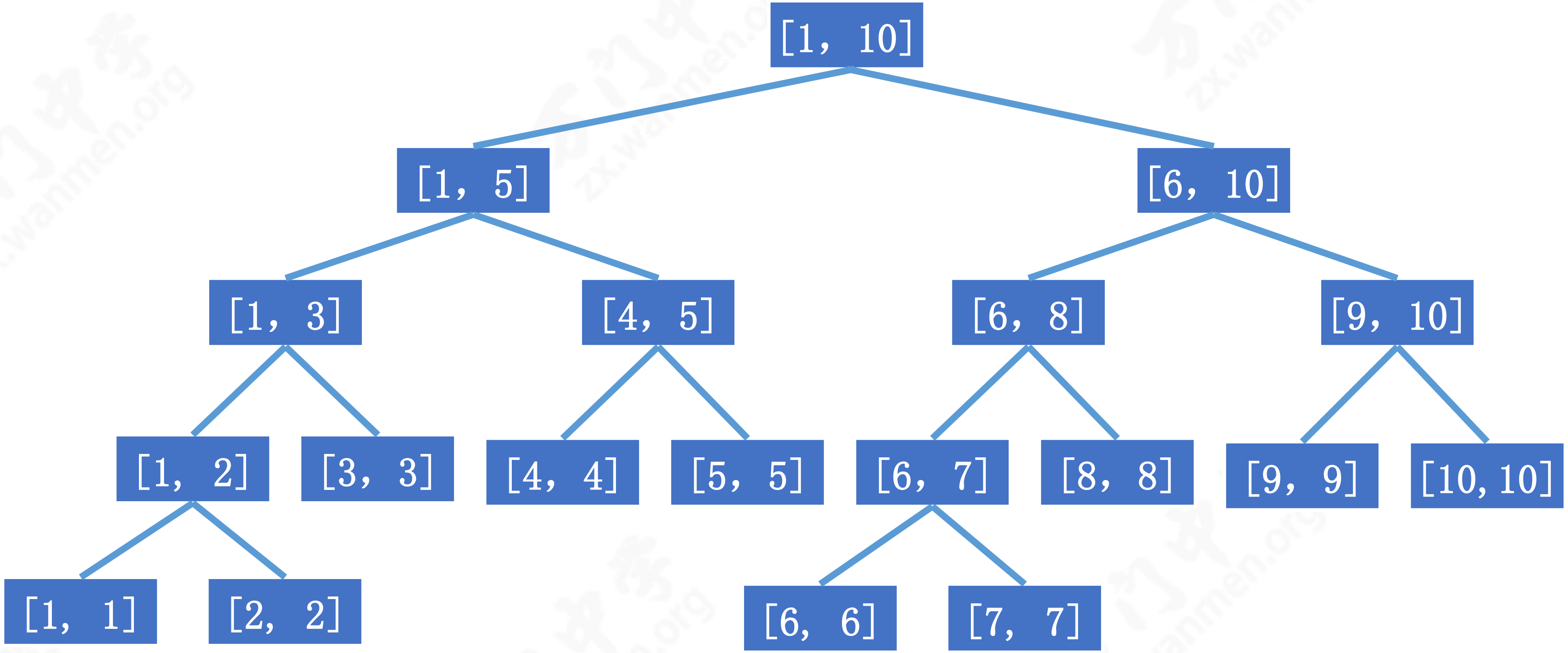
由此我们引入线段树，他可以将我们要处理的区间不相交地分成若干个小区间。

每次维护都可以在这样一些分解后的区间上进行。

每次查询的时候也可以根据这些分解后的区间信息合并起来得到答案。

线段树的结构

大小为 10 的线段树



线段树的结构

每个节点维护一个闭区间[l, r](l <= r)的信息。

根节点表示[1, n]的信息。

如果 l = r 那就是叶子结点.

如果 l < r 那就是内部节点，它有两个子节点 [l, (l+r)/2],

[(l+r)/2+1, r].

线段树的结构

用 1 表示根节点。

下标为 x 的点的左子节点下标为 2x，右子节点 2x+1。

用 sum[x] 表示 x 代表的区间里所有数的和。

对于叶子结点 x，它代表 [l, r](l = r)，sum[x] = a[l]

```
void update(int x){
	sum[x] = sum[x * 2] + sum[x * 2 + 1];
}
```

线段树的性质

结点数：

假设线段树处理的数列长度为 n，即根节点的区间为 [1, n]. 那么结点数不超过 2n 个.

因为 $n + n/2 + n/4 + n/8 + \cdots$，约为 2n 个。

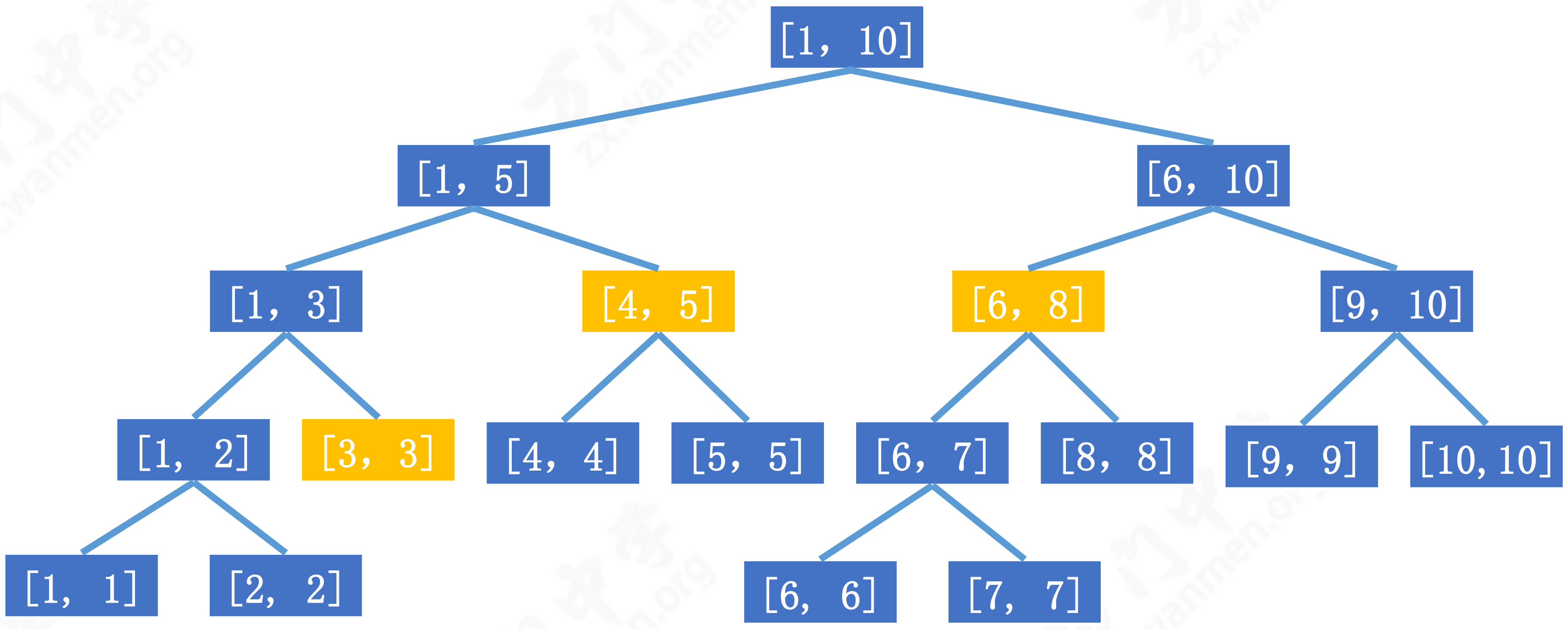
因此线段树的空间复杂度是 $O(n)$

深度：

可以看作满二叉树，深度不超过 $\log_2(n-1)+1$

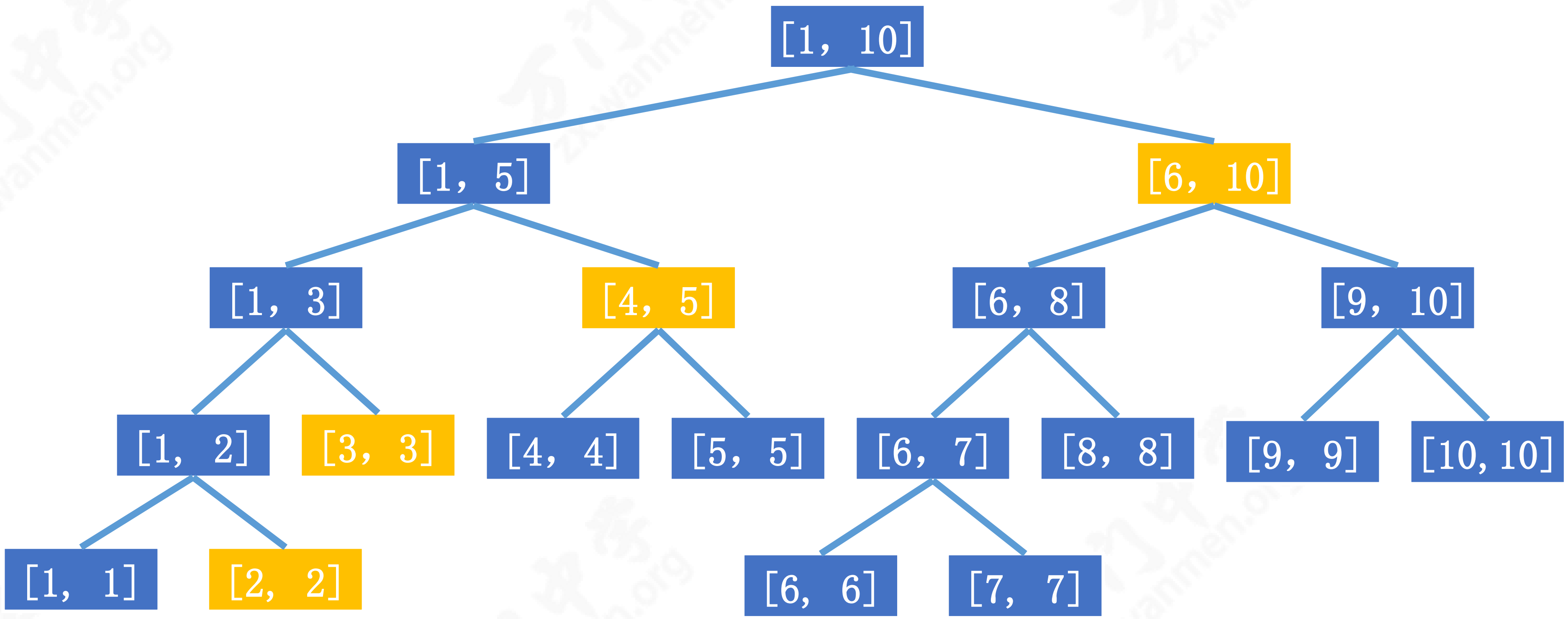
线段树上的区间划分

划分［3，8］这个区间。



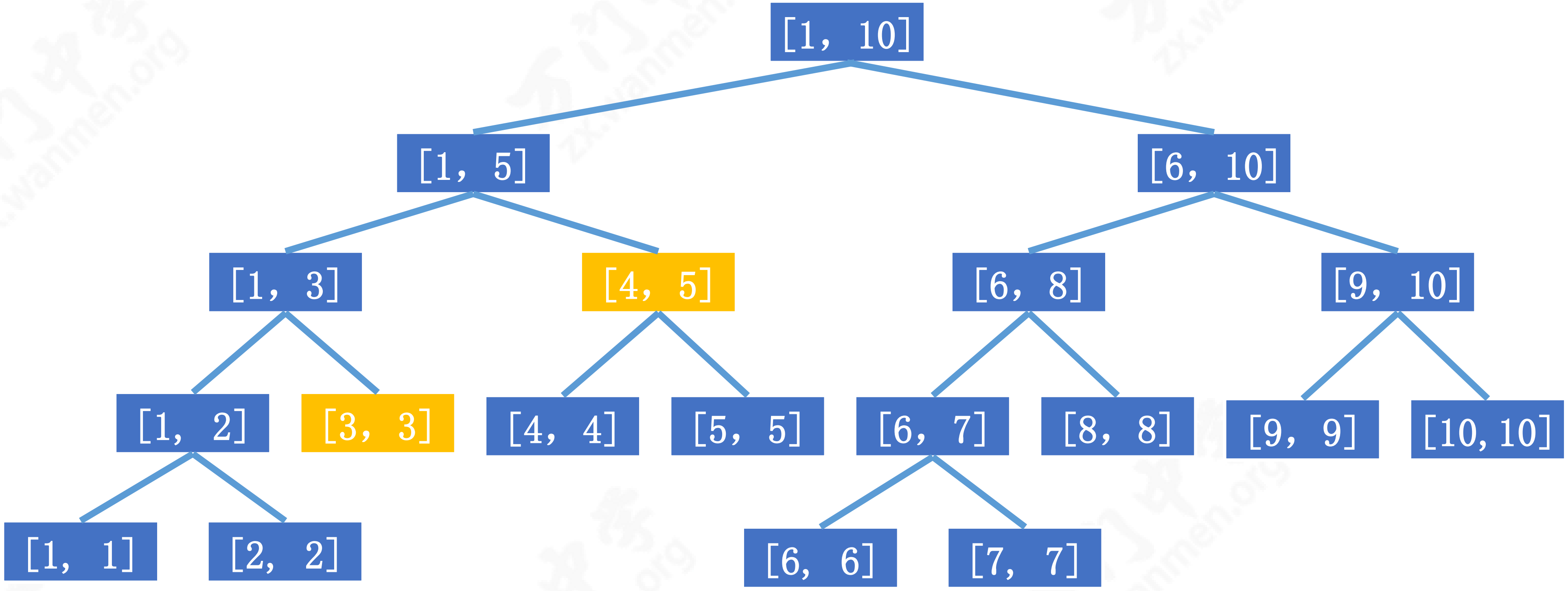
线段树上的区间划分

划分［2, 10］这个区间。



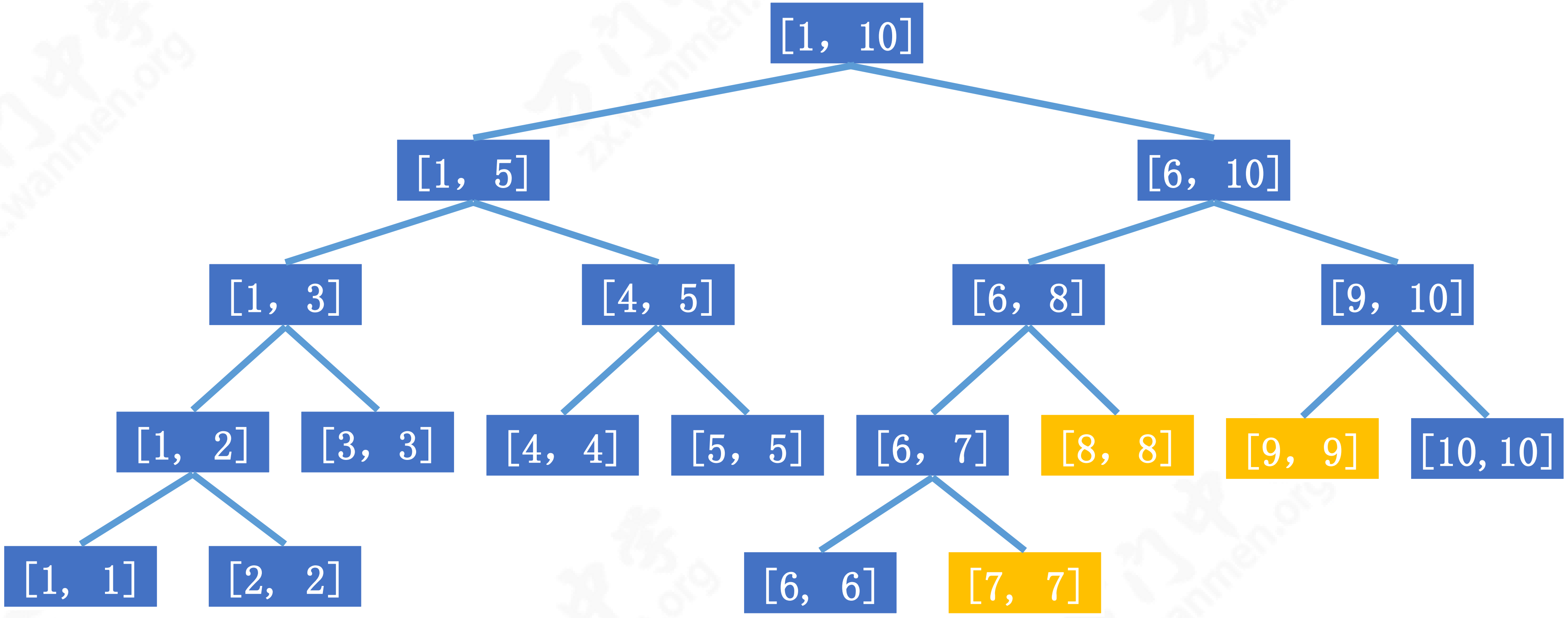
线段树上的区间划分

划分［3，5］这个区间。



线段树上的区间划分

划分［7，9］这个区间。



线段树的性质

线段树能把区间上的任意一条长度为 L 的线段都分成不超过 $2\log_2 L$ 条线段。

因此对于查询或修改某个长度为 L 的区间，我们只要在分解出来的线段上操作，然后在合并这几个区间的信息即可。

所以对于一般的操作，单次的时间复杂度都是 $O(\log n)$

下节课再见