线段树的单点修改

主讲人：邓哲也

线段树的修改

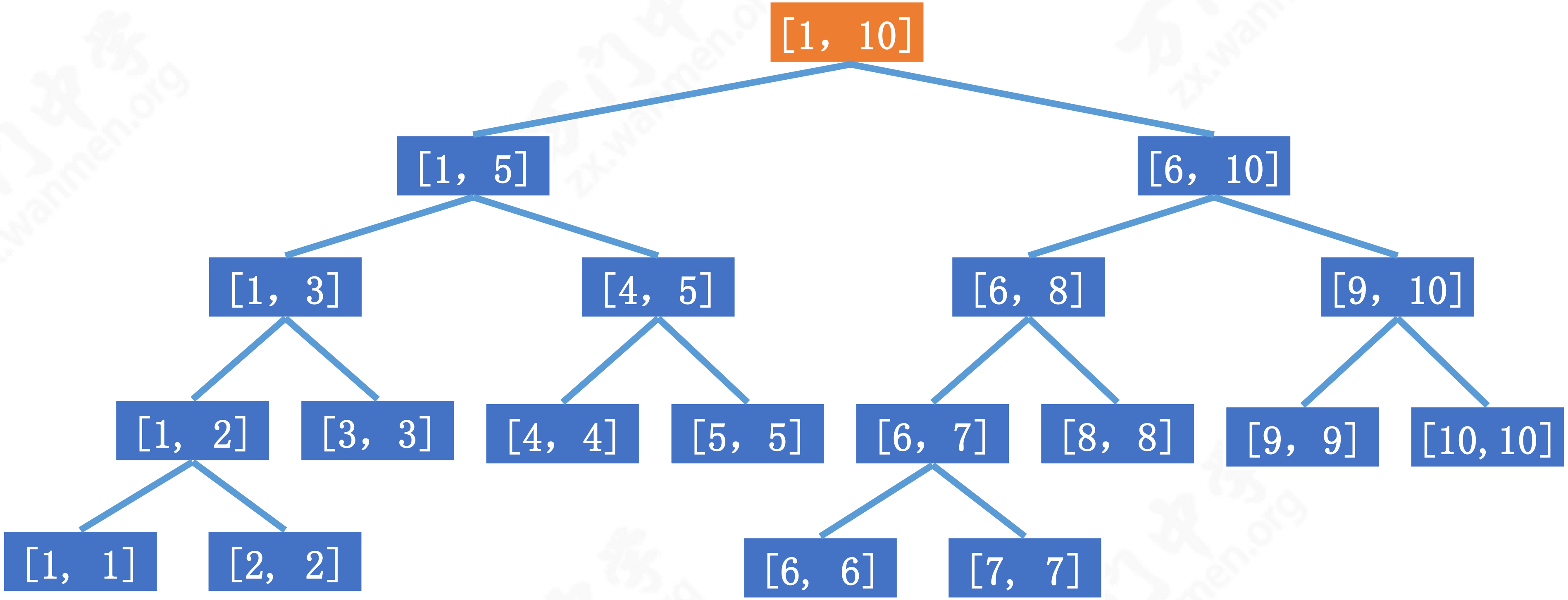
由于修改是对单个元素进行修改。

比如修改第 i 个元素。

我们先找到 [i, i] 所在的节点，然后修改它的 sum，然后一路向上更新每个祖先的 sum 即可。

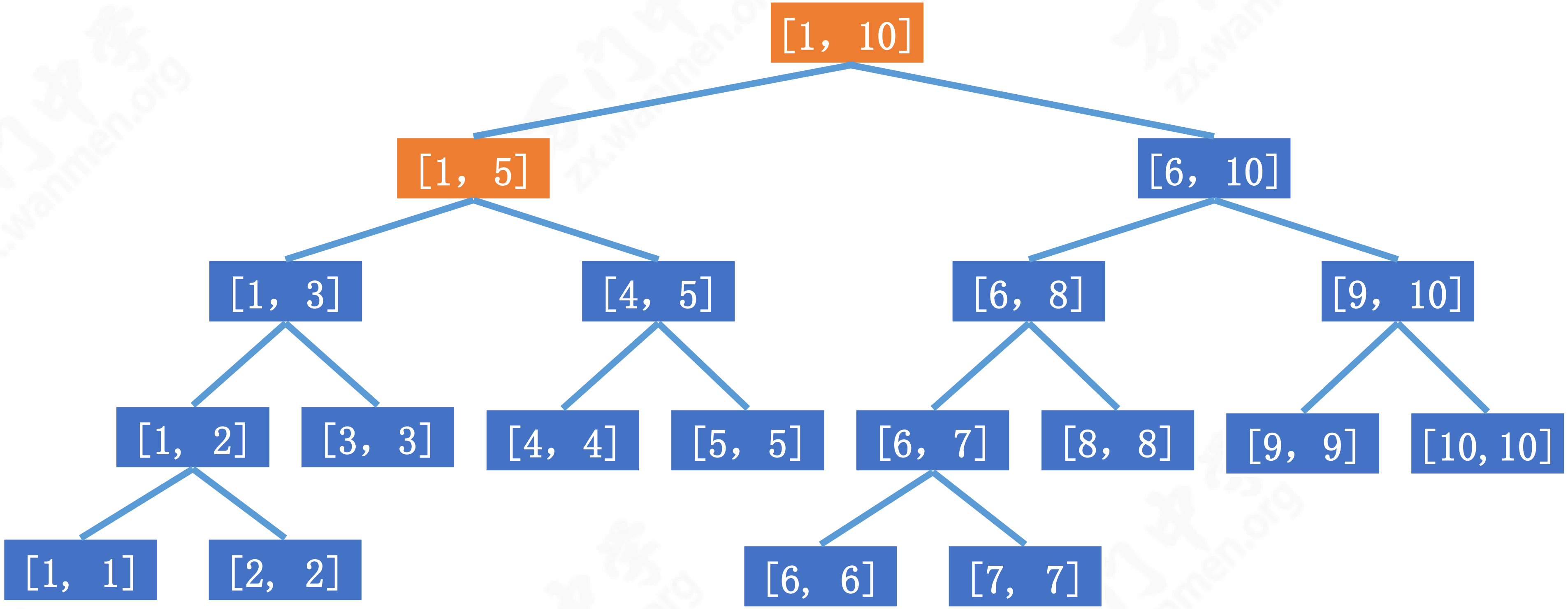
线段树的结构

修改 5 号元素的值。



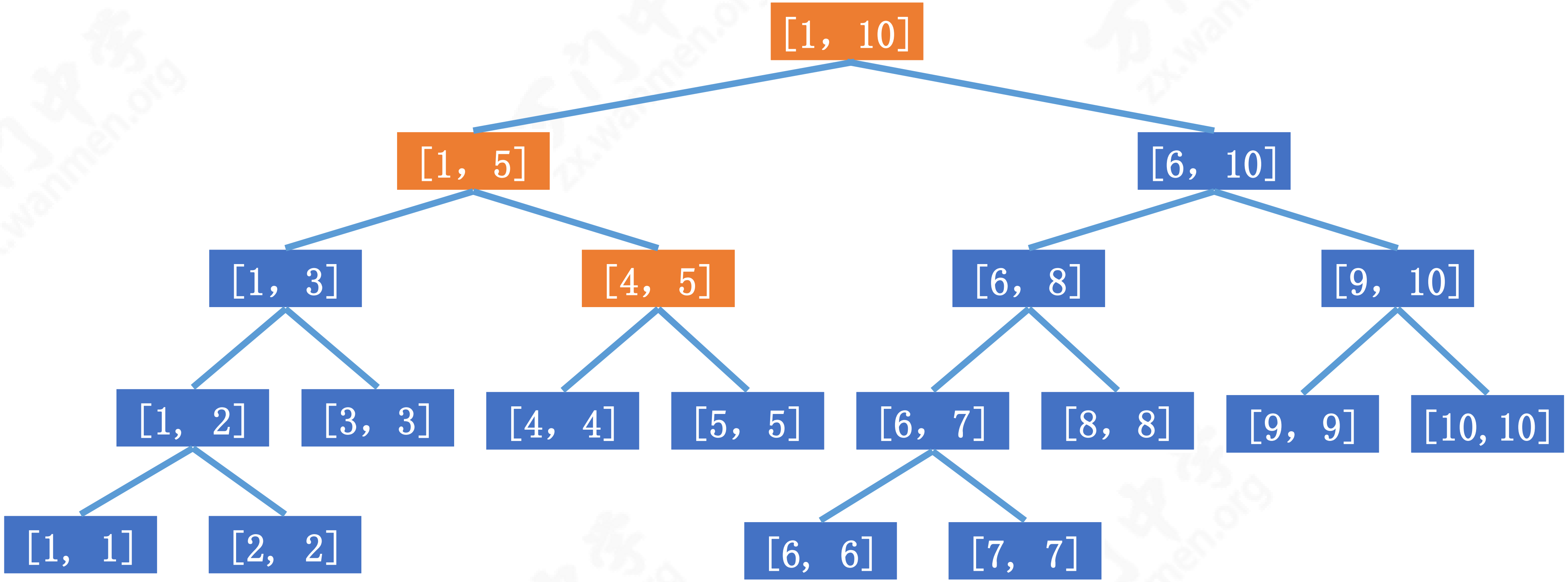
线段树的结构

修改 5 号元素的值。



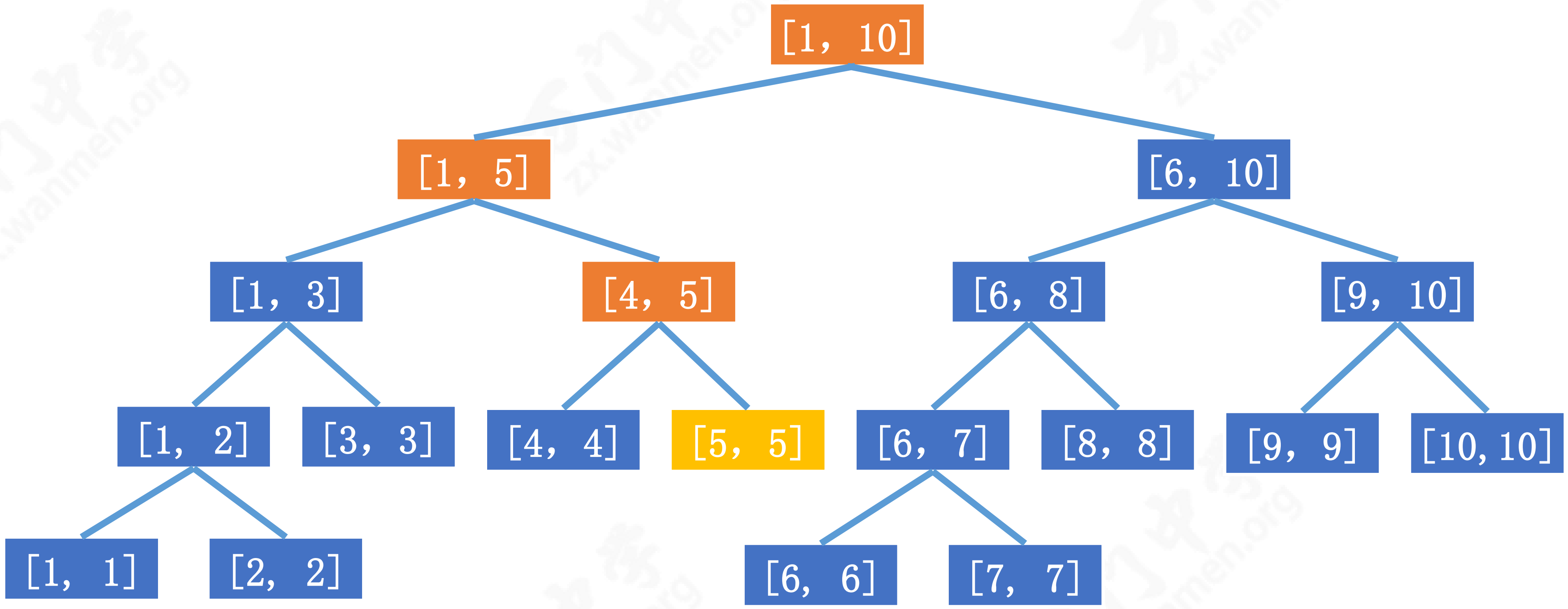
线段树的结构

修改 5 号元素的值。



线段树的结构

修改 5 号元素的值。



线段树的查询

```
int change(int pos, int v, int l, int r, int x){
        if (l == r){                            // 找到了要修改的叶子结点
                sum[x] = v;
                return;
        }
        int mid = (l + r) >> 1;
        if (pos <= mid)                                 // pos 在左子节点
                change(pos, v, l, mid, x * 2);
        else
                change(pos, v, mid + 1, r, x * 2 + 1);
        update(x);                      // 一定要加！因为这条路上的sum值发生了改变
}
```

总结

至此，我们已经解决了第一个问题，支持单点修改数字，和区间查询和。

两者的时间复杂度都是单次 $O(\log n)$。

边学边练

问题：给定一个序列，a[1]，a[2]，…，a[n]．求这个序列的逆序对数量。

逆序对是指一个有序二元组(i，j)，满足 i < j，a[i] > a[j].

$n <= 100000$，$0 <= a[i] <= 10^9$

样例：（输出 5）
5
2 7 9 6 4

边学边练

对于第 i 个数，我们要统计前面有多少个数大于 a[i]。

对每个数都统计一遍加起来即是答案。

假设我们可以对 [0, 10^9] 建一个线段树（实际上太大了）

每次先查询 [a[i]+1, 10^9] 的区间和，加入答案。

然后在 a[i] 的位置上加一即可。

边学边练

10^9范围太大了，因此我们先要对 n 个数进行离散化。

离散化的过程，就是对 n 个数进行排序，最小的数赋值为 1，第二小的赋值为 2，以此类推，这样 n 个数的取值范围就在［1，n］中了。

现在只要对［1，n］建立长度为 n 的线段树即可。

边学边练

离散化代码：　（假设对 a[1..n] 进行离散化）

```
int cnt = 0;
for (int i = 1;i <= n;i ++)
	bin[++ cnt] = a[i];
sort(bin + 1, bin + n + 1);
cnt = unique(bin + 1, bin + cnt + 1) - bin - 1;
for (int i = 1;i <= n;i ++)
	a[i] = lower_bound(bin + 1, bin + cnt + 1, a[i]) - bin;
```

边学边练

回到样例：2 7 9 6 4

现进行离散化：1 4 5 3 2

然后建立长度为 5 的线段树。

枚举到 1，查询[2，5]区间和为 0，a[1] ++

枚举到 4，查询[5，5]区间和为 0，a[4] ++

枚举到 5，查询[6，5]区间和为 0，a[5] ++

枚举到 3，查询[4，5]区间和为 2，a[3] ++

枚举到 2，查询[3，5]区间和为 3，a[2] ++

答案即为 0+0+0+2+3=5。

边学边练

对原始数组进行离散化，时间复杂度是O(n log n).

用线段树求解逆序对，枚举每个位置，做一次区间查询和一次单点修改，时间复杂度是O(n log n).

总的时间复杂度就是 O(n log n).

下节课再见