

主席树从入门到入门

by bLue

学长领进门
AC 靠个人

主席树常用来解决的问题

- 区间/树上路径第 K 大
- 区间不同数的个数

今天我们只讲

区间第 K 大

什么是主席树？



黄嘉泰

👛 渔业

🏠 清华大学 | 交叉信息研究院 | ♂

✓ 查看详细资料

+ 关注他

💬 发私信

动态

回答 20

提问 0

文章 0

专栏 0

想法 0

更多 ✓



该用户设置了隐私保护，登录查看他的动态、回答、文章等内容

个人成就

👍 获得 556 次赞同

获得 40 次感谢，68 次收藏

✎ 参与 15 次公共编辑

刘看山 · 知乎指南 · 知乎协议 · 应用 · 工作

侵权举报 · 网上有害信息举报专区

违法和不良信息举报：010-82716601

儿童色情信息举报专区

联系我们 © 2017 知乎



黄嘉泰 (HJT)

👜 渔业

🏠 清华大学 | 交叉信息研究院

▼ [查看详细资料](#)

FOTILE 主席

中国高端数据结构专家与领导者

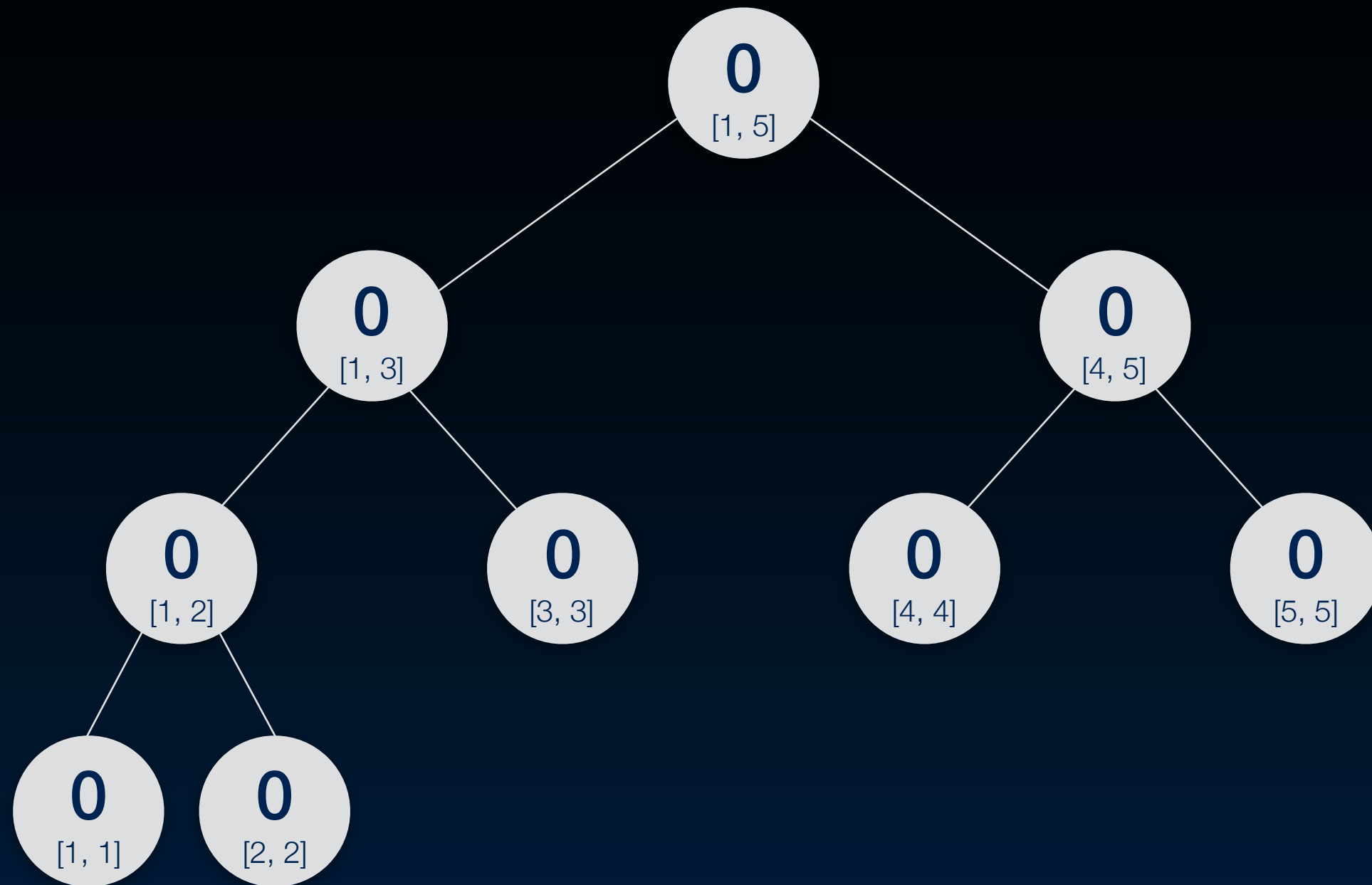
主席树的结构

- 由若干棵线段树组成（若是权值线段树通常需先对数据离散化）
- 按照输入顺序，每插入一个数保存一份当前版本的线段树

1 5 8 4 6

1 3 5 2 4

1 3 5 2 4



Day 1

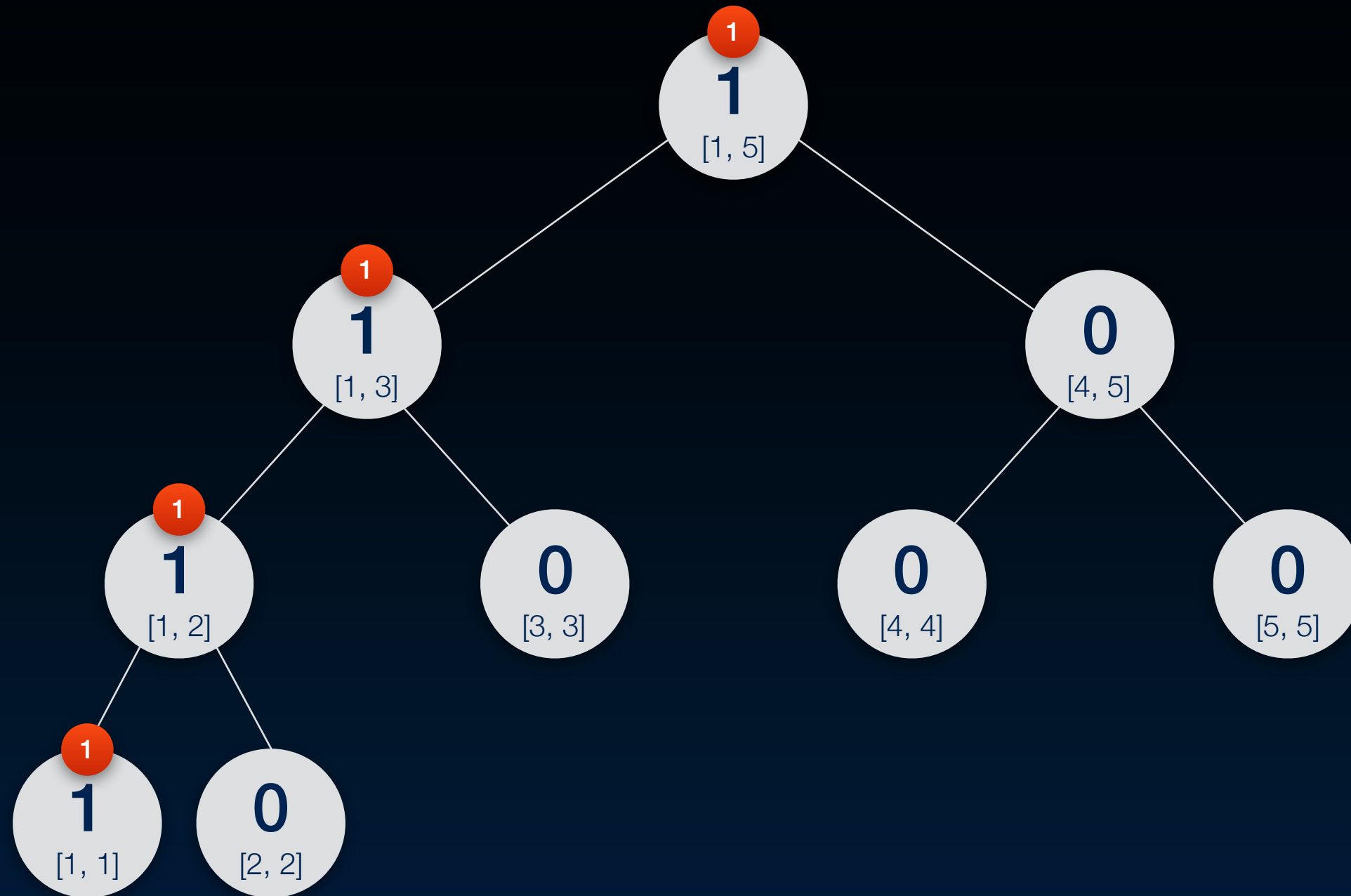
1

3

5

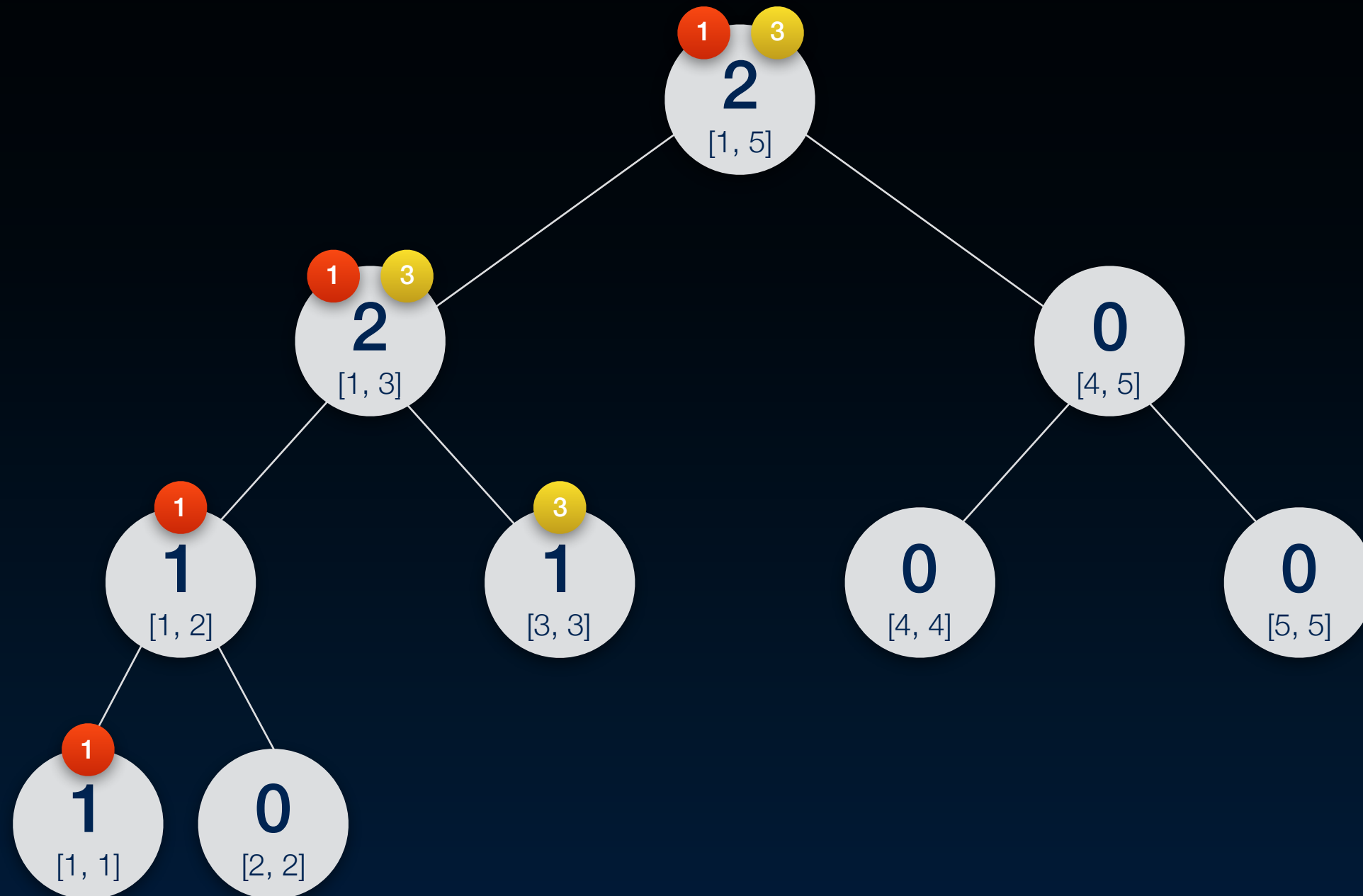
2

4



Day 2

1 3 5 2 4



Day 3

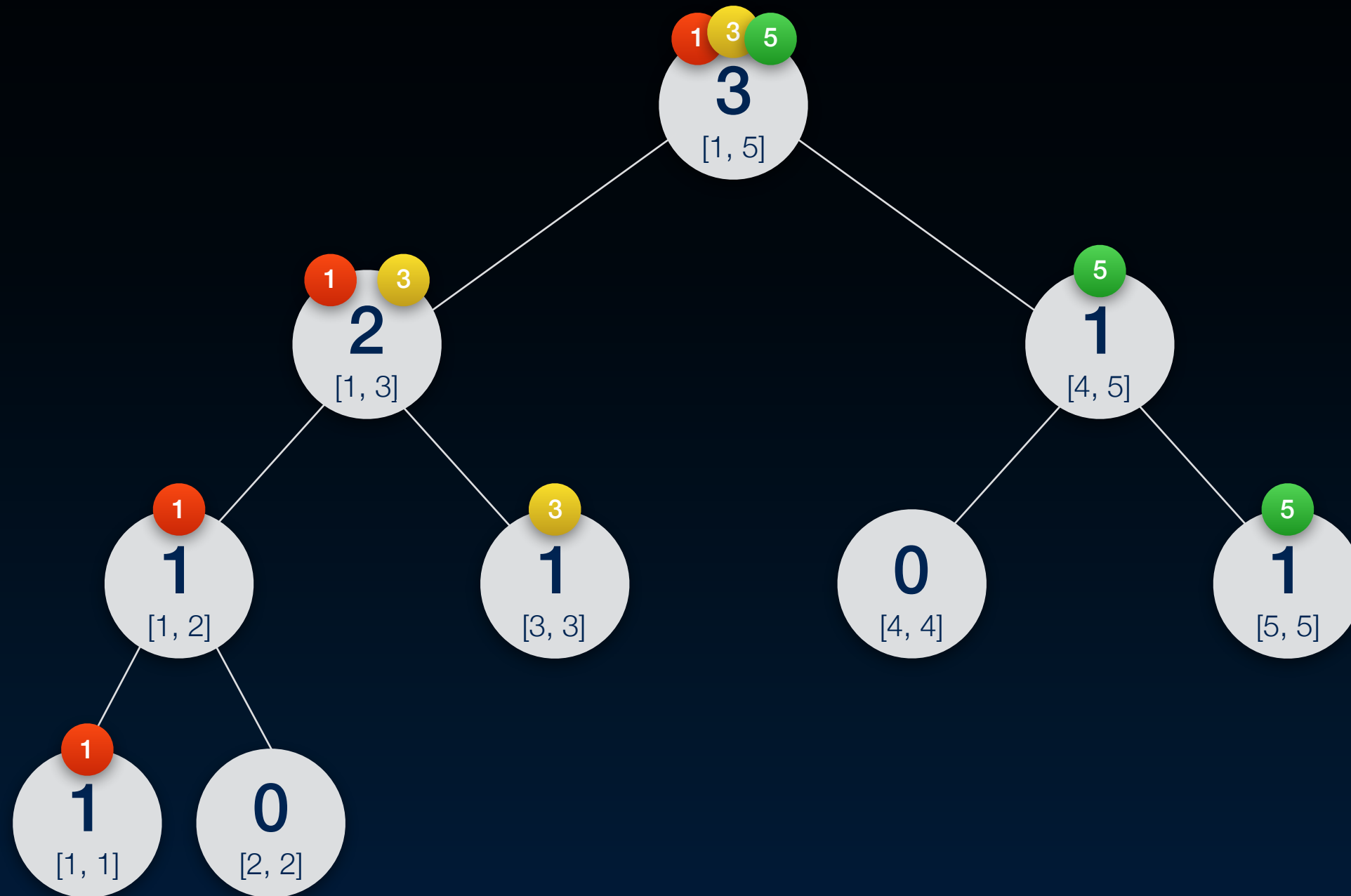
1

3

5

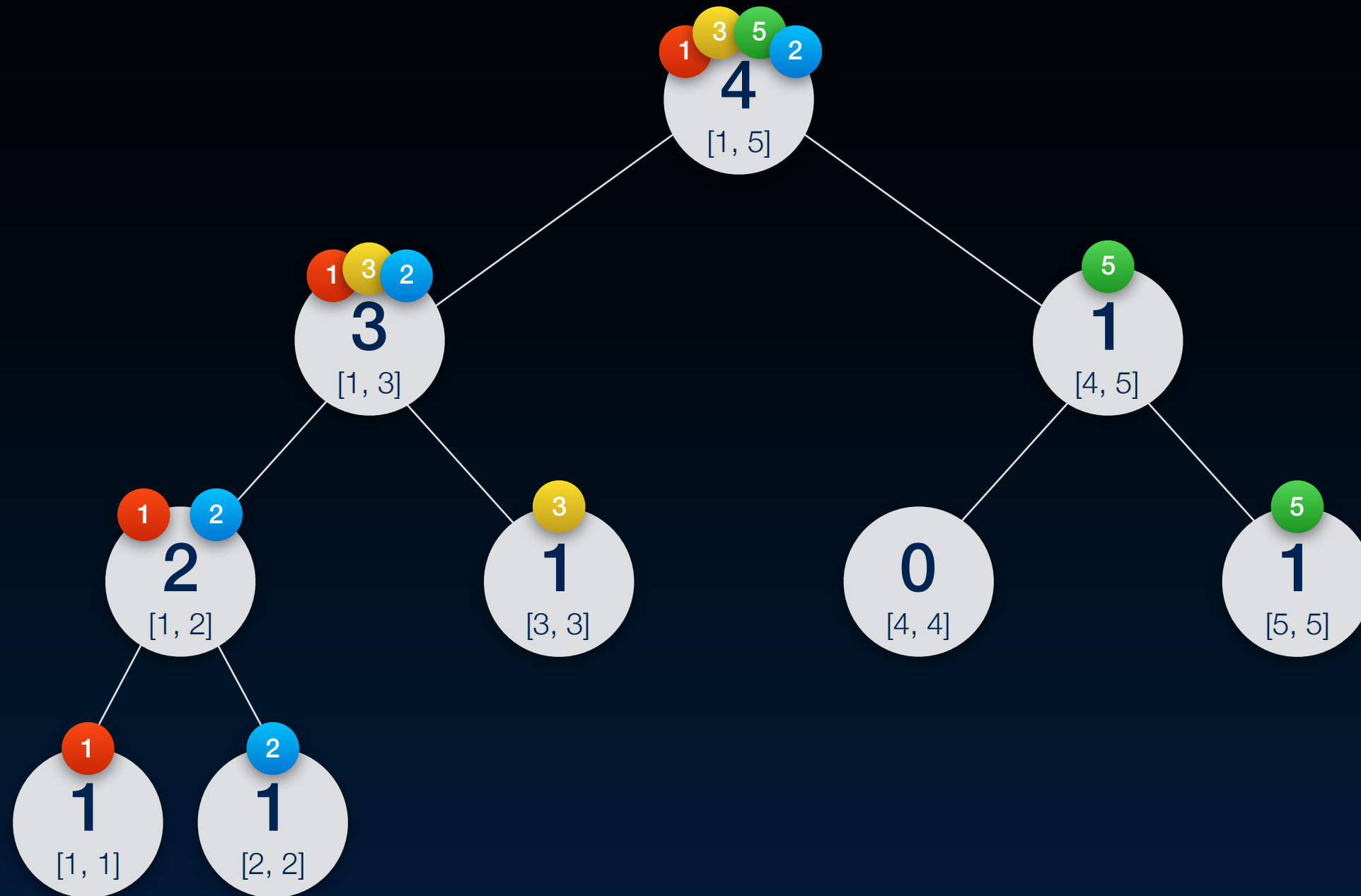
2

4



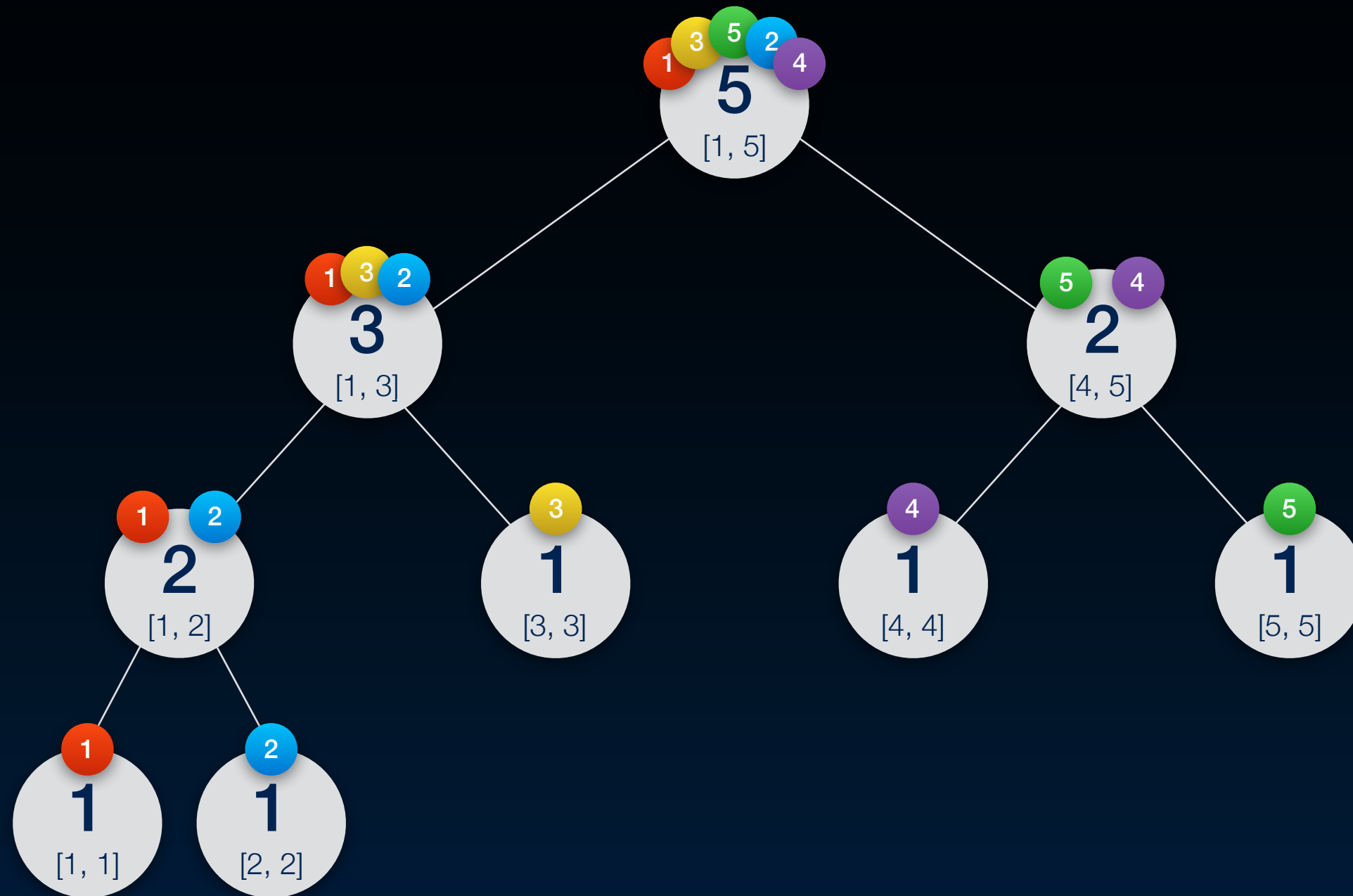
Day 4

1 3 5 2 4

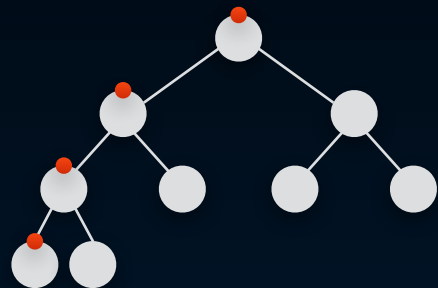


Day 5

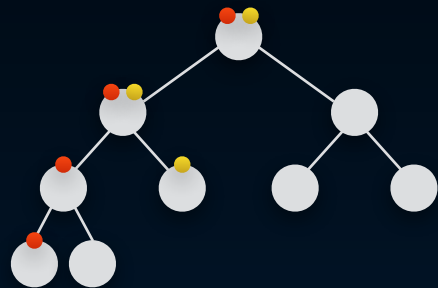
1 3 5 2 4



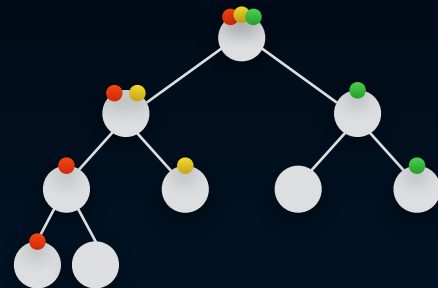
Day 1



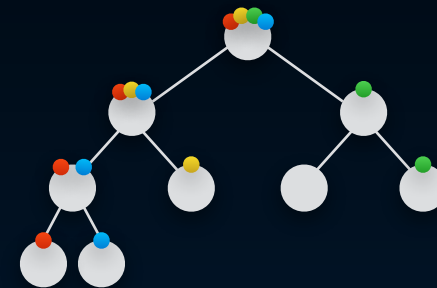
Day 2



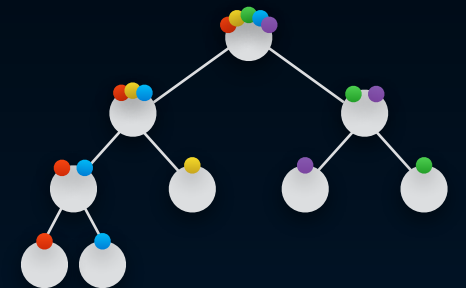
Day 3



Day 4



Day 5



如何查询区间第 K 大?

实例 (POJ 2104)

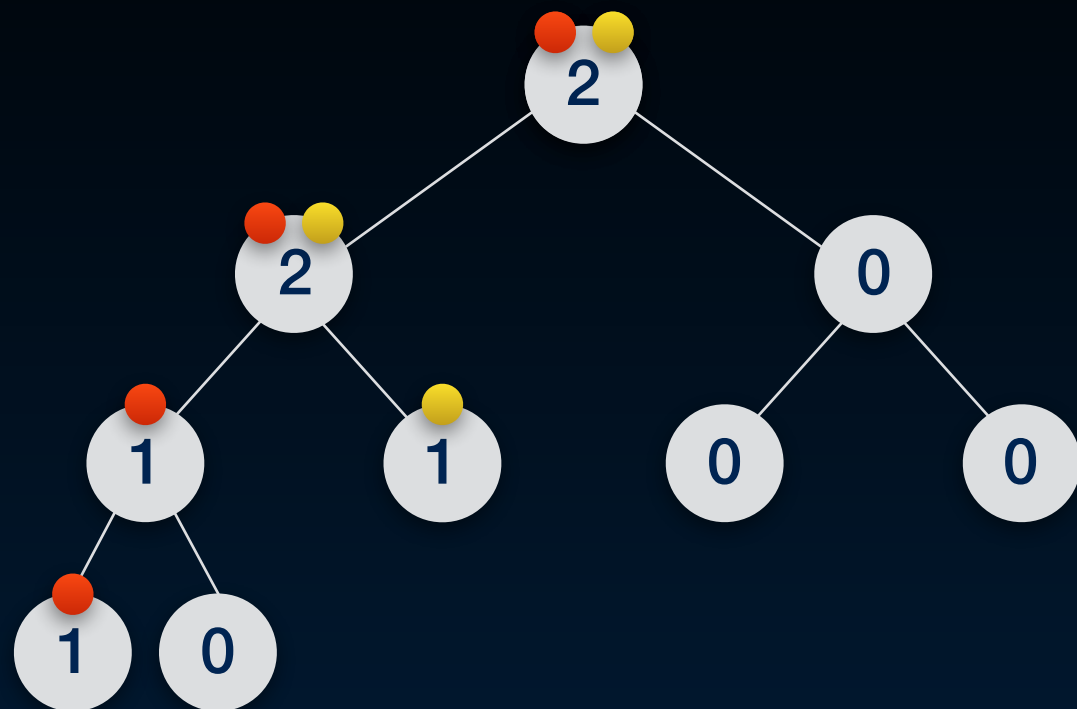
求区间 $[3, 5]$ 内第 2 小的数

实例 (POJ 2104)

求区间 $[3, 5]$ 内第 2 小的数

- 取第二天 $[1, 2]$ 和 第五天 $[1, 5]$ 的权值线段树
- 前缀和思想，将两颗线段树（对应结点）相减得区间 $[3, 5]$ 的线段树

Day 2



Day 5

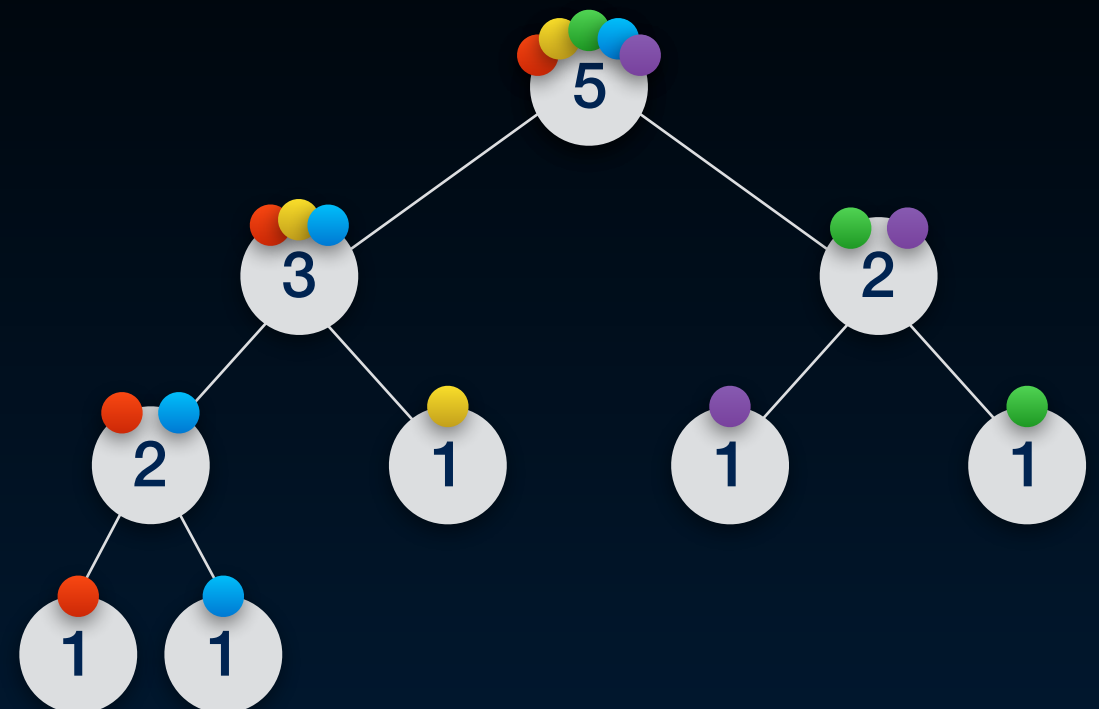


Diagram illustrating the subtraction of two sets:

Set 1 (Left): A circle labeled **5** with the interval $[1, 5]$ below it. It contains five colored dots labeled 1, 3, 5, 2, and 4.

Set 2 (Right): A circle labeled **2** with the interval $[1, 5]$ below it. It contains two colored dots labeled 1 and 3.

A minus sign ($-$) is placed between the two circles, indicating the operation of set subtraction.



The diagram illustrates the set difference operation $[1, 5] - [1, 5]$. It consists of three light gray circles arranged horizontally, separated by a minus sign and an equals sign.

- First Circle:** Contains the number **5** in the center and the interval $[1, 5]$ below it. Five colored circles are arranged in an arc above the number: red (1), yellow (3), green (5), blue (2), and purple (4).
- Second Circle:** Contains the number **2** in the center and the interval $[1, 5]$ below it. Two colored circles are arranged in an arc above the number: red (1) and yellow (3).
- Third Circle:** Contains the number **3** in the center and the interval $[1, 5]$ below it. Three colored circles are arranged in an arc above the number: green (5), blue (2), and purple (4).

The visual result shows that the elements 1 and 3 are removed from the first set, leaving the elements 5, 2, and 4.

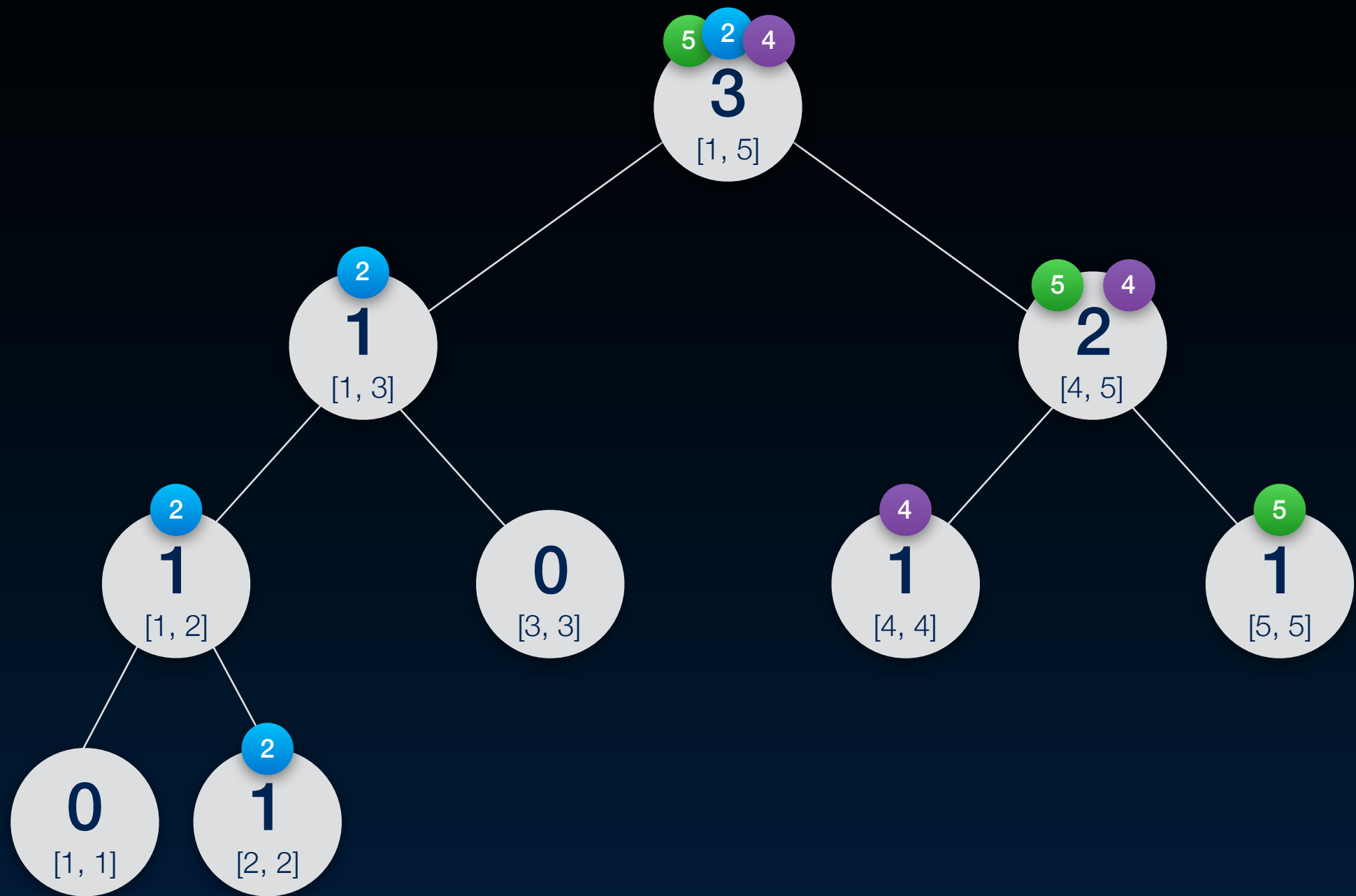
Day 3

+

Day 4

+

Day 5



实例 (POJ 2104)

求区间 $[3, 5]$ 内第 2 小的数

- 取第二天 $[1, 2]$ 和 第五天 $[1, 5]$ 的权值线段树
- 前缀和思想，将两颗线段树（对应结点）相减得区间 $[3, 5]$ 的线段树
- 在这棵线段树上查找第 K 小的数，方法为：
 - 若 $K \leq$ 左子树上元素数量 sum ，则递归到左子树找第 K 小
 - 否则递归到右子树找第 $K-sum$ 小

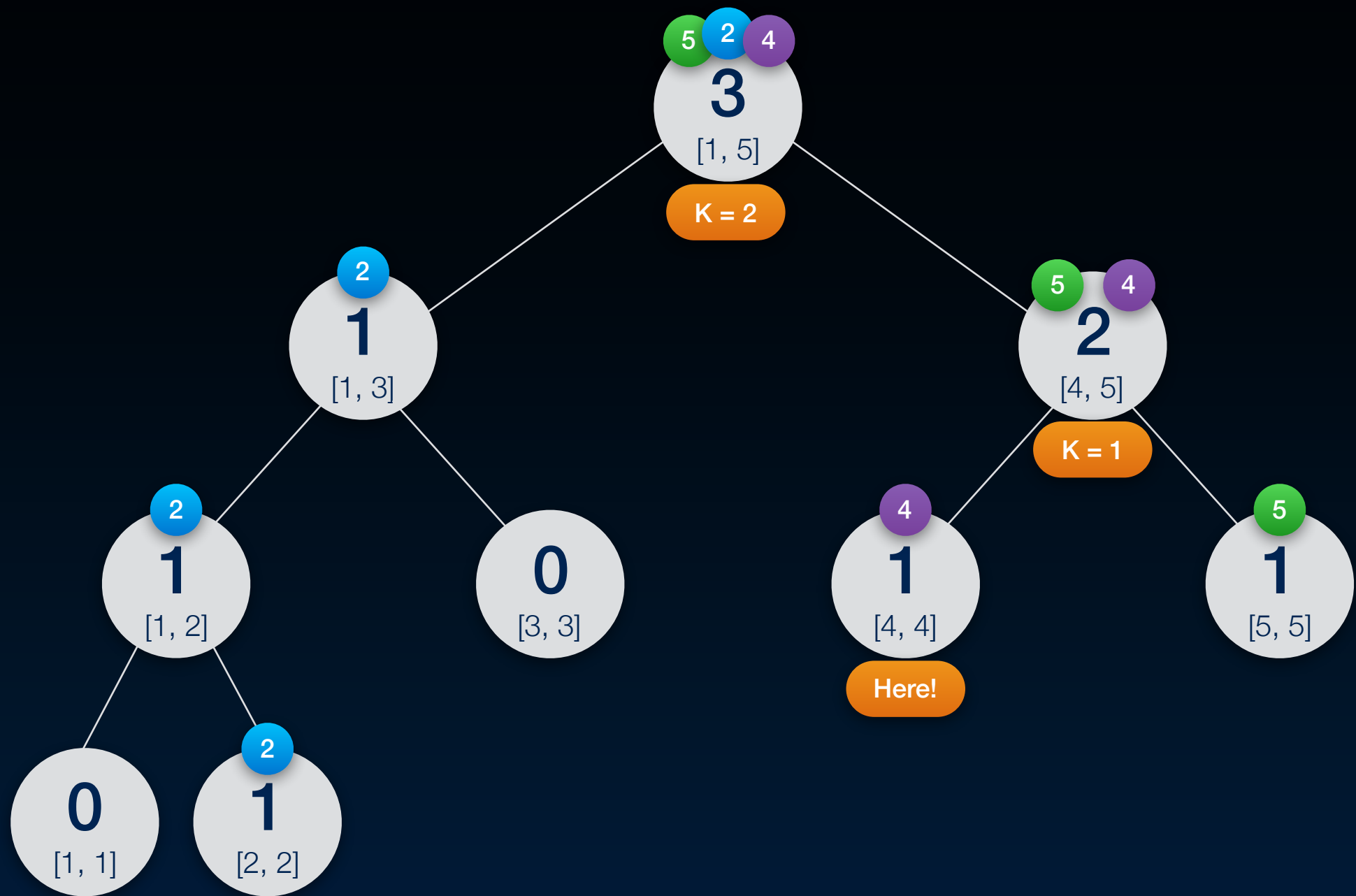
Day 3

+

Day 4

+

Day 5



实例 (POJ 2104)

求区间 $[3, 5]$ 内第 2 小的数

- 取第二天 $[1, 2]$ 和 第五天 $[1, 5]$ 的权值线段树
- 前缀和思想，将两颗线段树（对应结点）相减得区间 $[3, 5]$ 的线段树
- 在这棵线段树上查找第 K 小的数，方法为：
 - 若 $K \leq$ 左子树上元素数量 sum ，则递归到左子树找第 K 小
 - 否则递归到右子树找第 $K-sum$ 小

现在就来大胆地 1A 吧！

[Submit](#)[☆Favorite](#)[Submissions](#)[Leaderboard](#)

Time limit	20000 ms
Case time limit	2000 ms
Memory limit	65536 kB
OS	Linux
Source	Northeastern Europe 2004 , Northern Subregion

Descriptions:

[System Crawler](#) 2017-09-20[teamcww](#) 2016-08-17

K-th Number [POJ - 2104](#)

You are working for Macrohard company in data structures department. After failing your previous task about key insertion you were asked to write a new data structure that would be able to return quickly k-th order statistics in the array segment.

That is, given an array $a[1..n]$ of different integer numbers, your program must answer a series of questions $Q(i, j, k)$ in the form: "What would be the k-th number in $a[i..j]$ segment, if this segment was sorted?"

For example, consider the array $a = (1, 5, 2, 6, 3, 7, 4)$. Let the question be $Q(2, 5, 3)$. The segment $a[2..5]$ is $(5, 2, 6, 3)$. If we sort this segment, we get $(2, 3, 5, 6)$, the third number is 5, and therefore the answer to the question is 5.

Input

The first line of the input file contains n --- the size of the array, and m --- the number of questions to answer ($1 \leq n \leq 100\,000$, $1 \leq m \leq 5\,000$).

The second line contains n different integer numbers not exceeding 10^9 by their absolute values --- the array for which the answers should be given.

The following m lines contain question descriptions, each description consists of three numbers: i, j , and k ($1 \leq i \leq j \leq n$, $1 \leq k \leq j - i + 1$) and represents the question $Q(i, j, k)$.

Output

For each question output the answer to it --- the k-th number in sorted $a[i..j]$ segment.

Sample Input

```
7 3
```



测试弹幕

+1s

你退群吧

怎么突然有人了

我从未见过有如此厚颜无耻之人

退款退款!

我真是哔了bLue了

←前面说退款的你交钱了吗?

我学了假的主席树

emmmmmm...

UP主翻车了

喵喵喵喵喵?

#10782226 | 1A_ZhuXiShu's solution for [POJ-2104]

flag

交错题就神作了

乌鸦坐飞机

这个怕是翻车了2333

控制不当 被水淹没

Status

Length

Lang

Submitted

Shared

RemoteRunId

我可能学了假算法

flag

BGM: 聋的传人

Memory Limit Exceeded

1527

G++

2017-09-20 21:06:16

17644925

我从未见过有如此厚颜无耻之人

主席看了想打人

Select Code

MLE_kenan: 喵喵喵?

求BGM

告诉大家一个秘密: 其实学

```
#include <cstdio>
```

```
#include <algorithm>
```

假算法+1

```
using namespace std;
```

MLE? 你在逗我? ? ?

给跪了

```
const int MAXN = 100001;
```

说好的 1A 呢? 假算法+10086

```
struct node {
```

苟

```
int sum, l, r;
```

```
} *hjt;
```

我怕是真的409

造成这种现象的主要原因是什么？

- A. 控制不当
- B. 玄学
- C. RP 问题
- D. 无可奉告
- E.  爆了内存 $O(n^2)$

动态开点

- 每次插入仅新建有修改的这一条链上的结点 ($\log n$ 个)
- 其他结点用指针指向前一棵线段树上的结点
- 空间复杂度下降到 $O(n \log n)$

总结

- 基础结构：线段树
- 可持久化：每次插入保存一个新版本，利用共用数据减少时空消耗
- 动态开点，节省内存
- 时空复杂度 $O(n \log n)$


```

// 本代码中所有数据均按从下标 1 开始存放

// 主席树中的线段树结点, sum 表示此区间内元素个数
struct node {
    int sum, l, r;
} hjt[MAXN*40];

int a[MAXN], sorted[MAXN], num; // sorted: 离散化后的数组 num: 离散化后的数组长度
int root[MAXN], cnt; // root: 主席树中用来保存每棵线段树树根的数组 cnt: 线段树结点数 (用于数组方式动态开点)

// 查找离散化之后的下标
int GetIdx(int v) {
    return lower_bound(sorted+1, sorted+1+num, v) - sorted;
}

// 初始化
void Init() {
    cnt = 0;
    root[0] = 0;
}

// 创建结点
inline int CreateNode(int sum, int l, int r) {
    int idx = ++cnt;
    hjt[idx].sum = sum;
    hjt[idx].l = l;
    hjt[idx].r = r;

    return idx;
}

// 新建一棵线段树, 只沿更新路径新建出较上个版本有修改的结点
// 调用参数
// root: 插入后新生成的线段树的根结点会赋值到 root 中存储
// pre_rt: 上一棵线段树的根
// pos: 本次要插入的数在线段树中的位置
// l, r: 递归参数。默认填写 1, num
void Insert(int &root, int pre_rt, int pos, int l, int r) {
    // 动态创建结点, 直接根据上一个版本复制对应的节点, sum+1
    root = CreateNode(hjt[pre_rt].sum+1, hjt[pre_rt].l, hjt[pre_rt].r);
    if(l == r) return;
    int m = (l+r) >> 1;
    if(pos <= m)
        Insert(hjt[root].l, hjt[pre_rt].l, pos, l, m);
    else Insert(hjt[root].r, hjt[pre_rt].r, pos, m+1, r);
}

// 本函数适用于查询区间 [l, r] 中的第 k 小。通常需要自行变通
// 调用参数
// s, e: 要查询区间所需的两个线段树的根, 如要查询区间 [l, r], 则传入 root[l-1], root[r]
// l, r: 递归参数。默认填写 1, num
// k: 要查询区间第几小
int Query(int s, int e, int l, int r, int k) {
    if(l == r) return l;
    int m = (l+r) >> 1;
    int sum = hjt[hjt[e].l].sum - hjt[hjt[s].l].sum; // 计算左子树的元素数量
    if(k <= sum) // 如果 k <= sum, 则 k 在左子树, 否则在右子树
        return Query(hjt[s].l, hjt[e].l, l, m, k);
    else return Query(hjt[s].r, hjt[e].r, m+1, r, k-sum);
}

```

习题

- POJ 2104: 区间第 K 大
- SPOJ DQUERY: 区间不同数的个数
- SPOJ COT: 树上路径第 K 大
- ZOJ 2112: 区间第 K 大 (带单点修改)

<https://vjudge.net/contest/186451>

