07-B

搜索树应用

线段树

把一条线分割成不相等的两段，再把这两段按照同样的比例再分成两个部分。假设第一次分出来的两段中，一段代表可见世界，另一段代表理智世界。然后再看第二次分成的两段，他们分别代表清楚与不清楚的程度，你便会发现，可见世界那一段的第一部分是它的影像

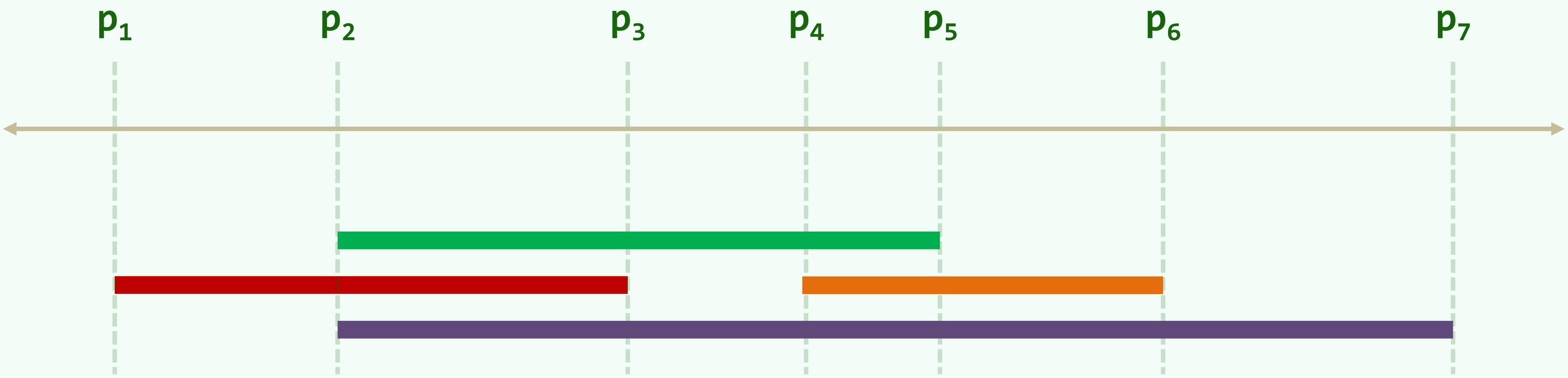
邓俊辉

deng@tsinghua.edu.cn

基本区间/Elementary Interval

❖ 任给x-轴上的n段区间：$I = \{ [x_i, x_i'] \mid i = 1, 2, 3, \ldots, n \}$

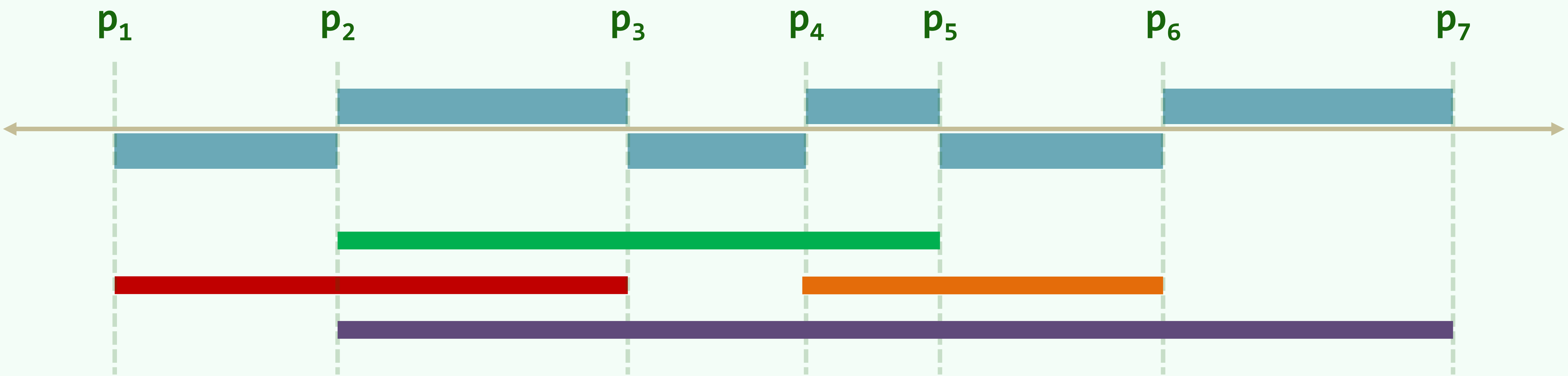
❖ 将它们的端点排序为：$\{ p_1, p_2, p_3, \ldots, p_m \}, \ m \leq 2n$



❖ 于是，便得到**m+1段基本区间/EI**：$(-\infty, p_1], \ (p_1, p_2], \ (p_2, p_3], \ \ldots, \ (p_{m-1}, p_m], \ (p_m, +\infty]$

离散化/Discretization

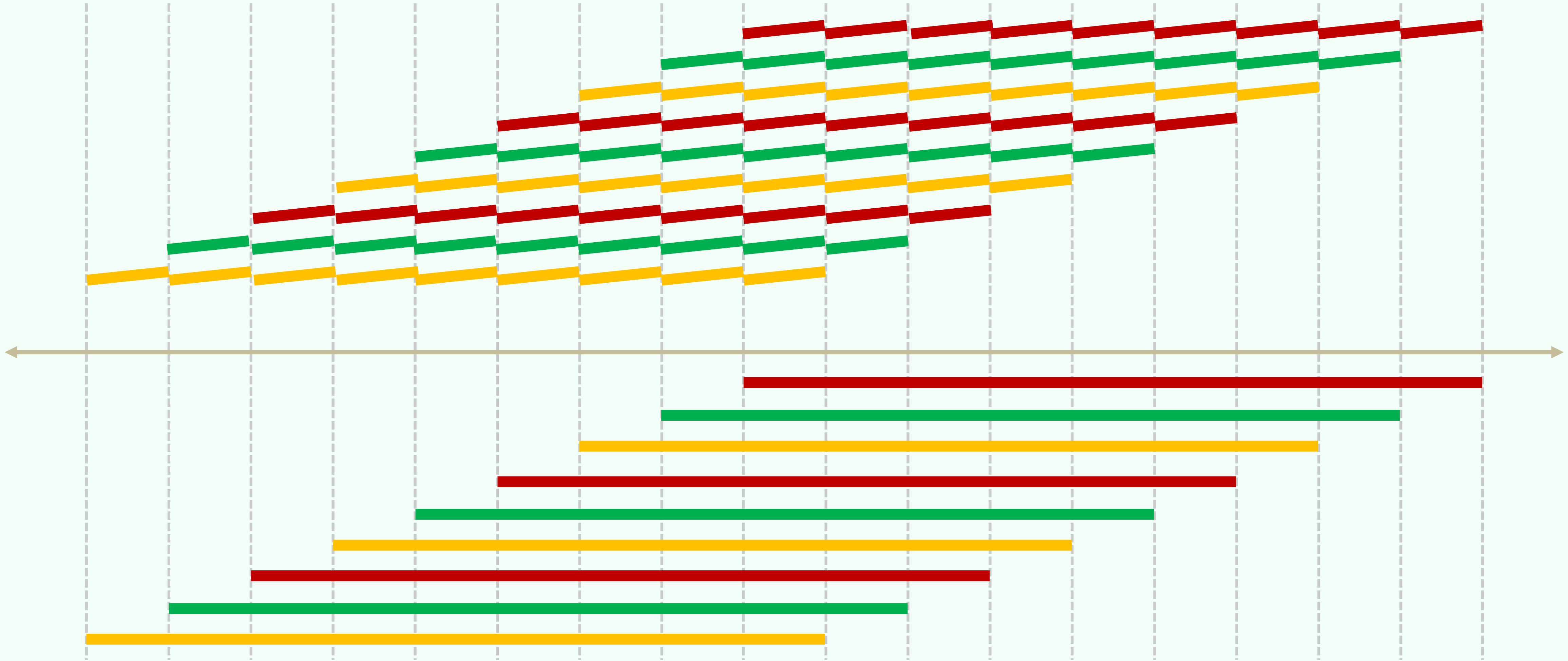
- 观察：同一段EI内的任何位置，穿刺查询的输出必然完全一样
- 于是，只要将所有所有EI预处理为有序向量，并使每段EI记录其对应的查询输出，那么...



- ...一旦确定穿刺位置，便可快速地完成查询：二分查找 + 输出结果 //$\mathcal{O}(\log n + r)$

空间爆炸

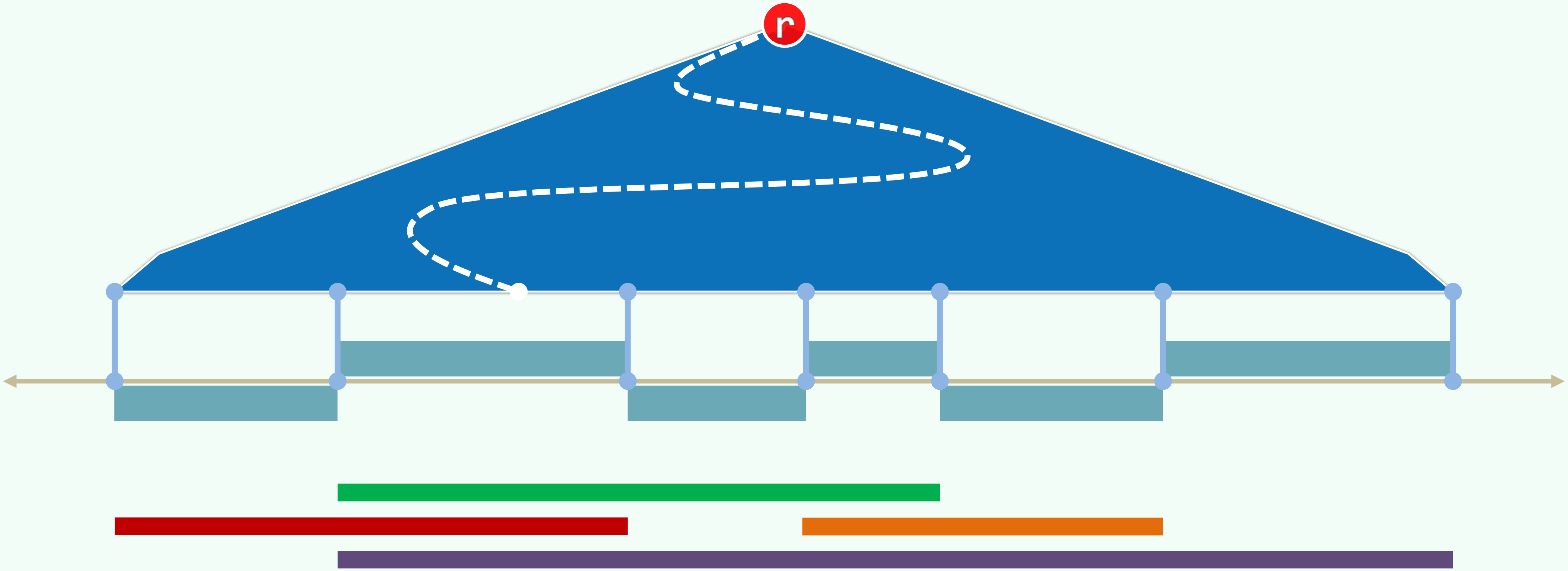
❖ 在最坏情况下，输入的每段区间都会横跨$\Omega(n)$段EI，总共需要$\Omega(n^2)$附加空间



将有序向量升级为BBST

❖ 节点v的**作用域**R(v)：叶节点v的R(v)，就是其对应的EI；内部节点v的R(v)，为其孩子作用域之**并**

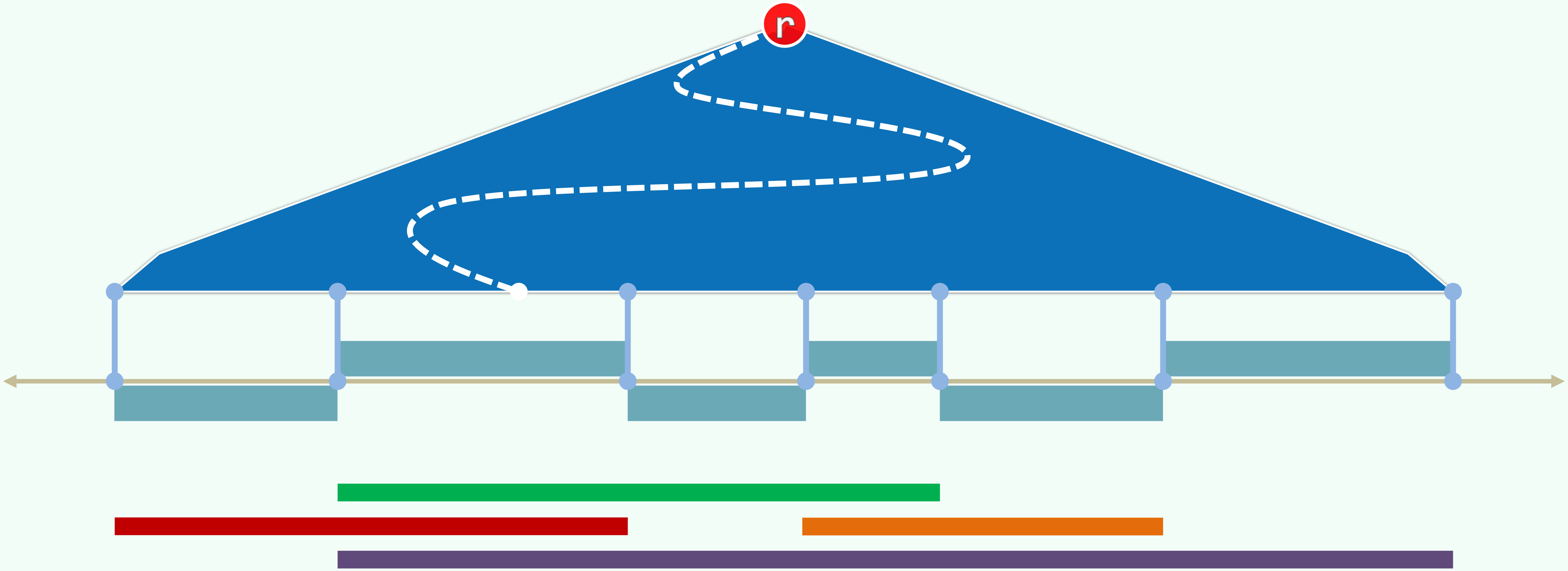
❖ 每个叶节点v各自预先记录**Int**(v)：即横跨R(v)的所有输入区间，作为该EI对应的查询输出



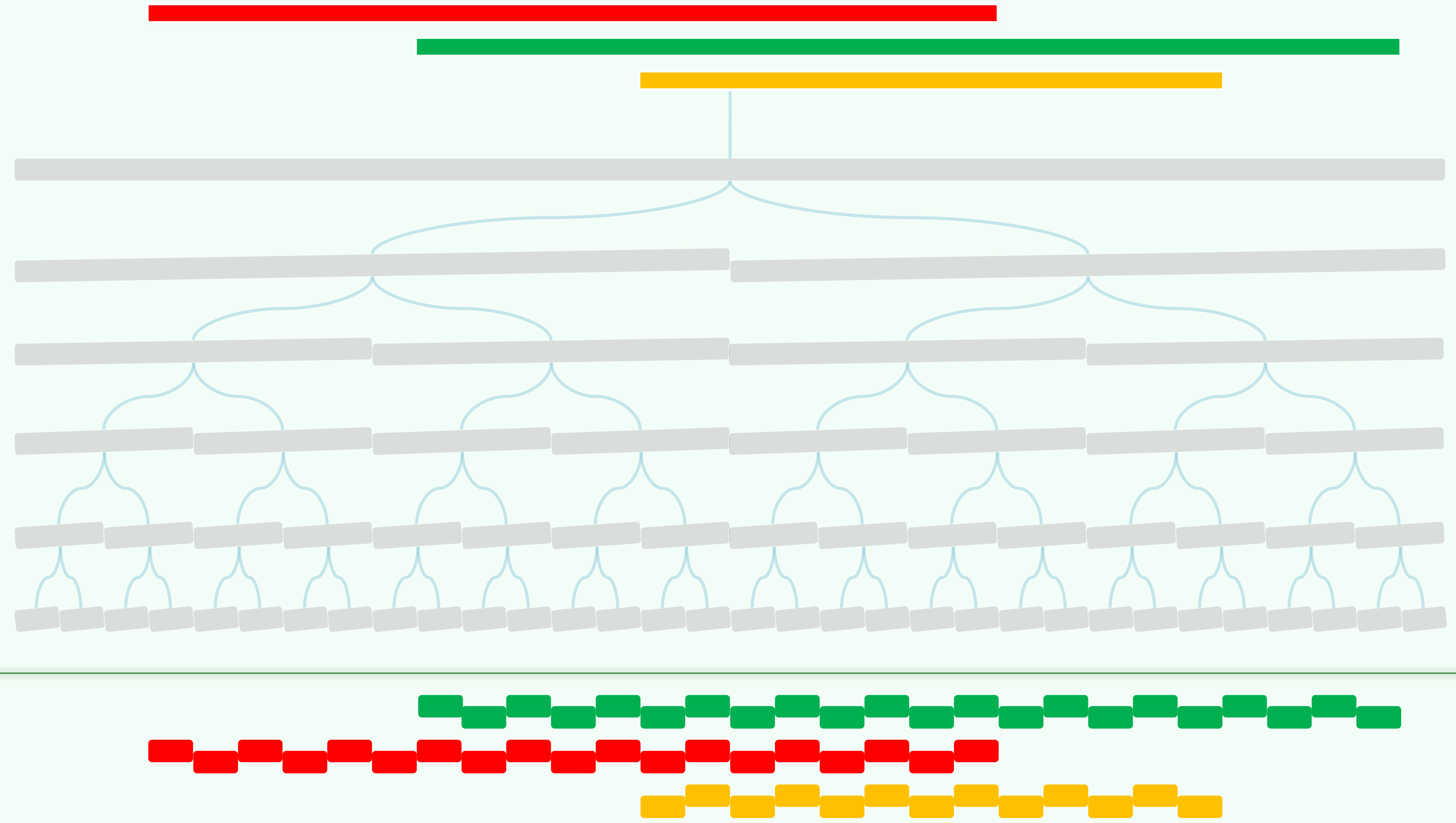
支持穿刺查询

❖ 对任何穿刺位置q_x，只需

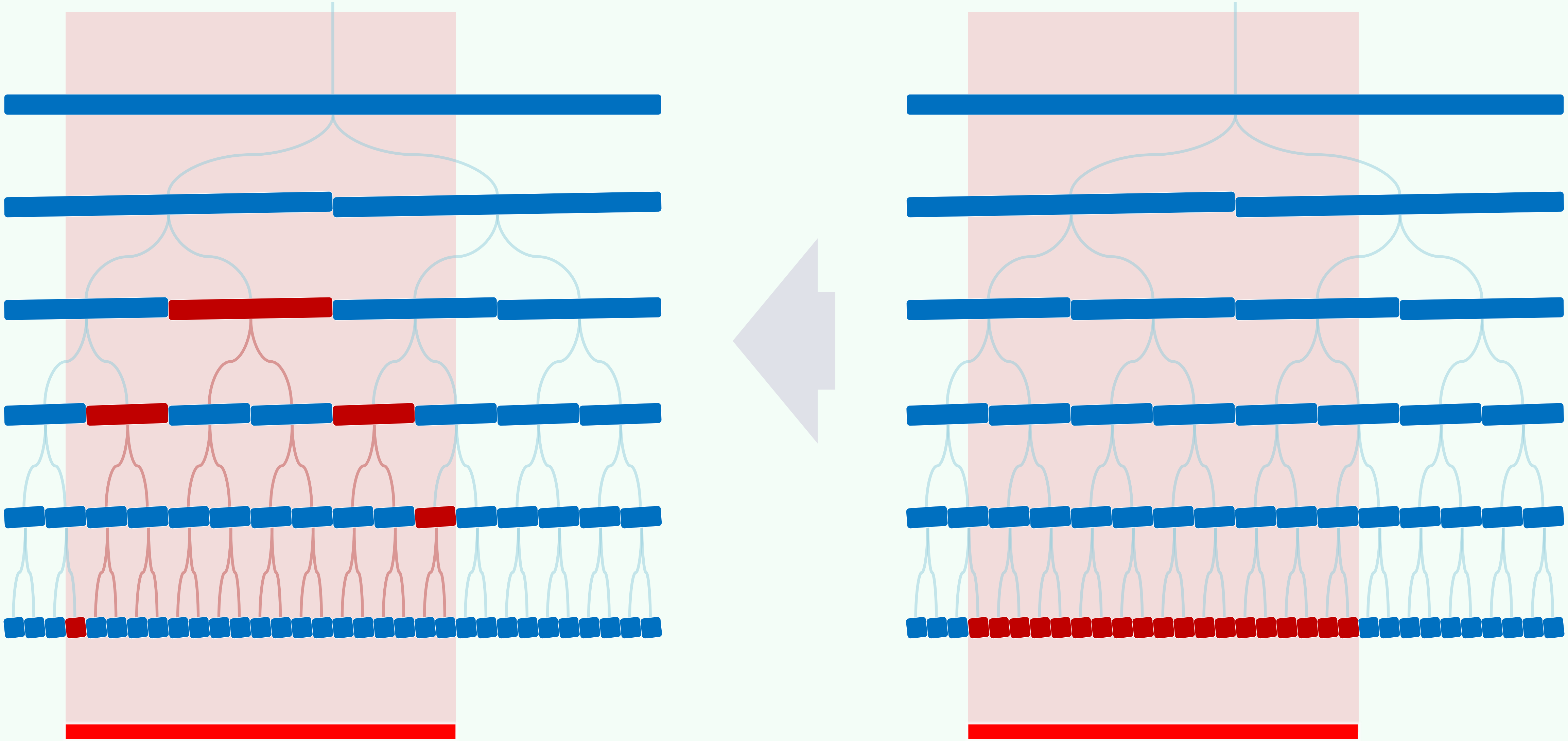
通过BBST::search(q_x)找到对应的叶节点v，并输出预先记录的集合Int(v) //$\mathcal{O}(\log n + r)$



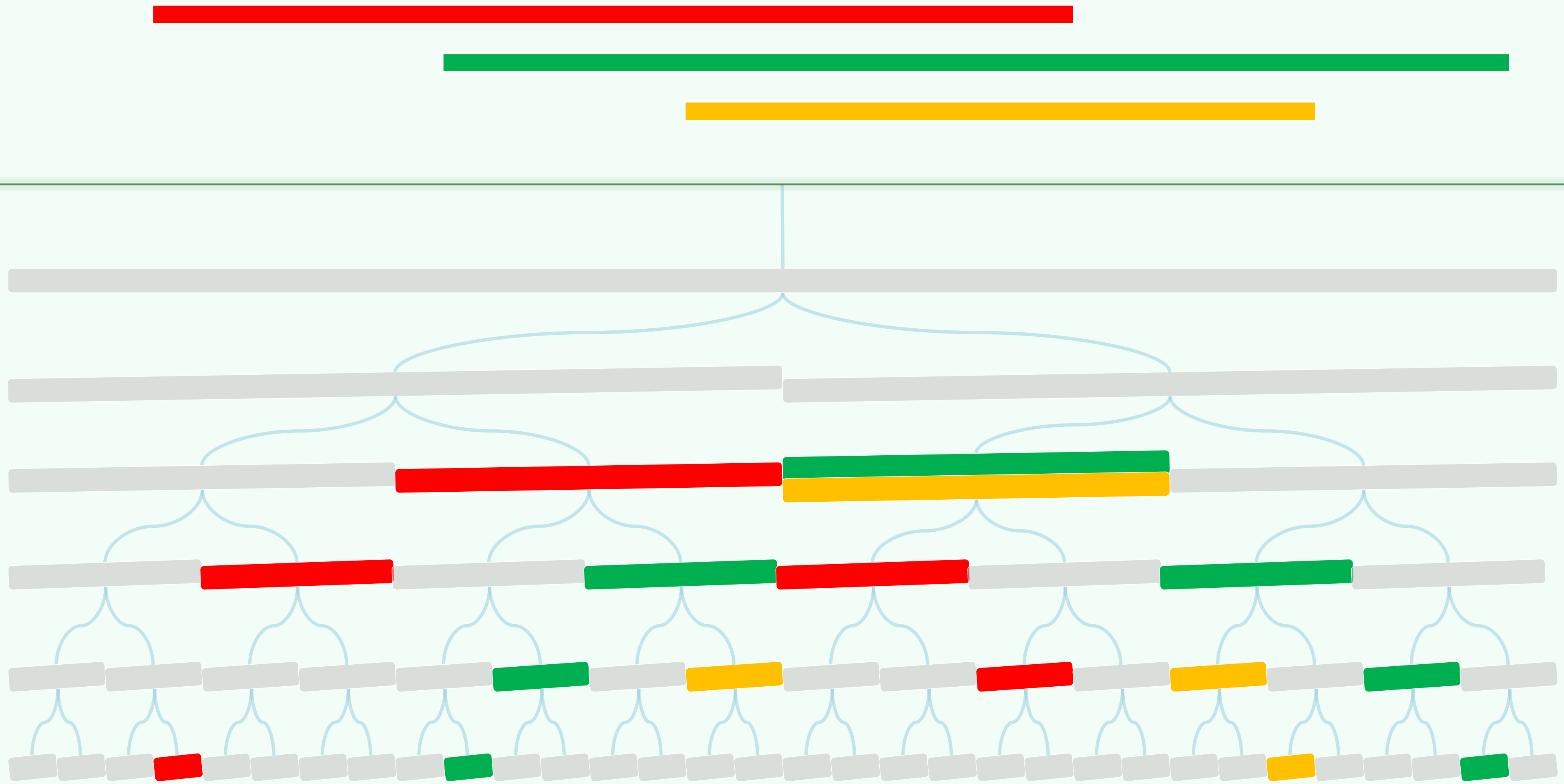
然而，最坏情况下仍需Ω(n^2)



通过贪心地合并，使每段输入区间至多被$O(\log n)$个祖先记录



从而，空间复杂度降至$O(n\log n)$



构造算法：BuildSegTree(I)

对I中所有区间的端点**排序** //$\mathcal{O}$(nlogn)

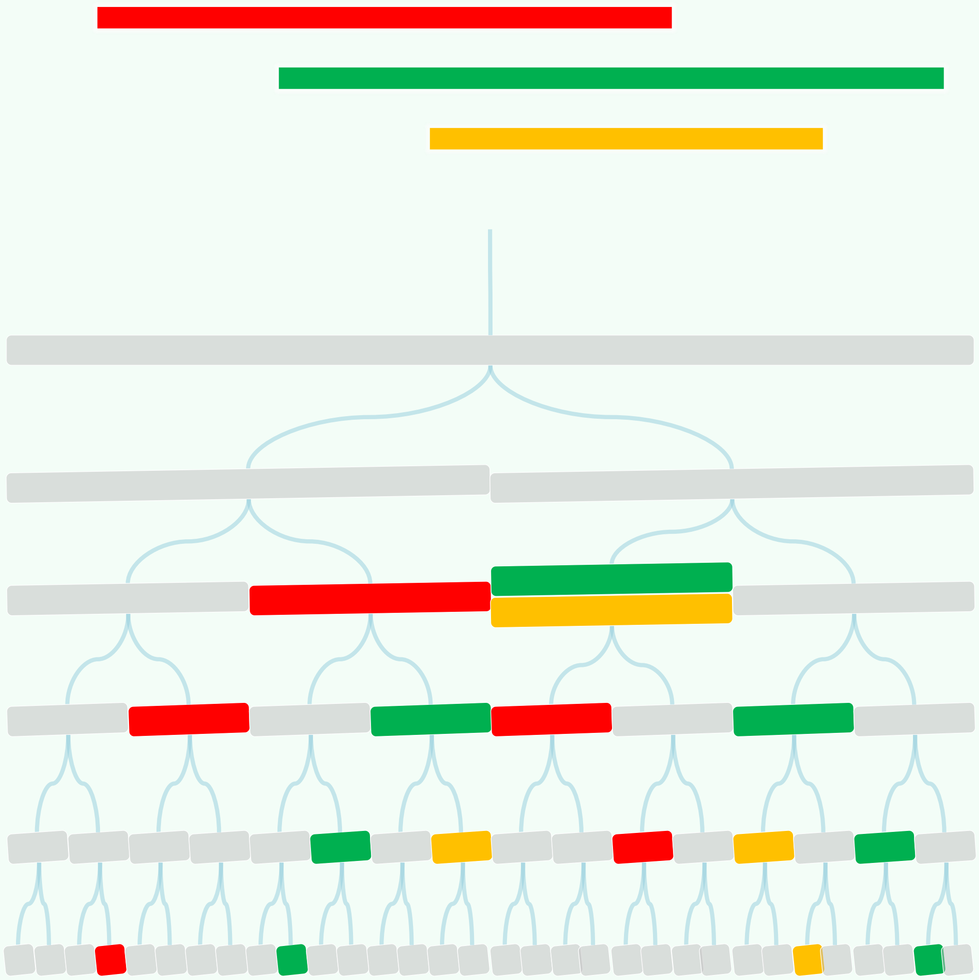
确定所有的EI

并将它们组织为一棵**BBST** //$\mathcal{O}$(n)

自底而上，确定

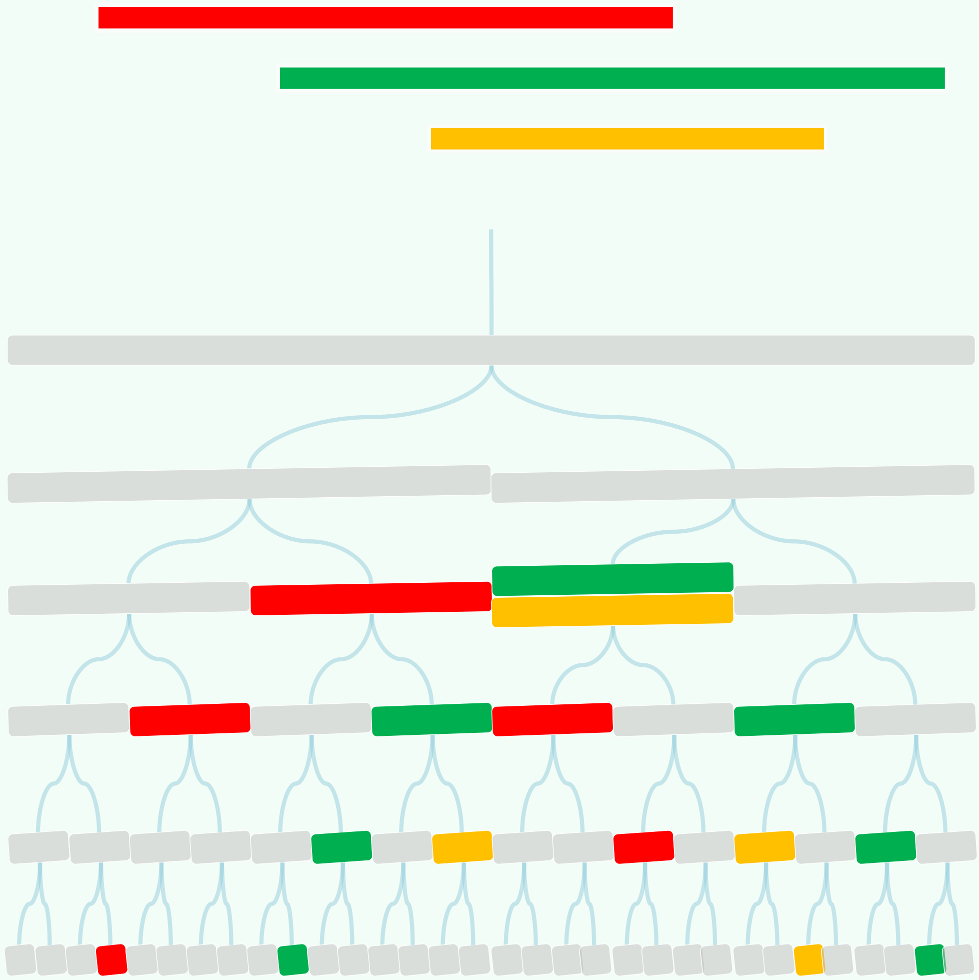
每个节点v的作用域**R**(v) //$\mathcal{O}$(n)

```
for ( I中的每一段区间s )
    InsertSeg( T.root, s )
```



构造算法：InsertSeg(v, s)

```
if ( R(v) ⊆ s ) //greedy by top-down
   store s at v and return;

if ( R( lc(v) ) ∩ s ≠ ∅ ) //recurse
   InsertSeg( lc(v), s );

if ( R( rc(v) ) ∩ s ≠ ∅ ) //recurse
   InsertSeg( rc(v), s );
```



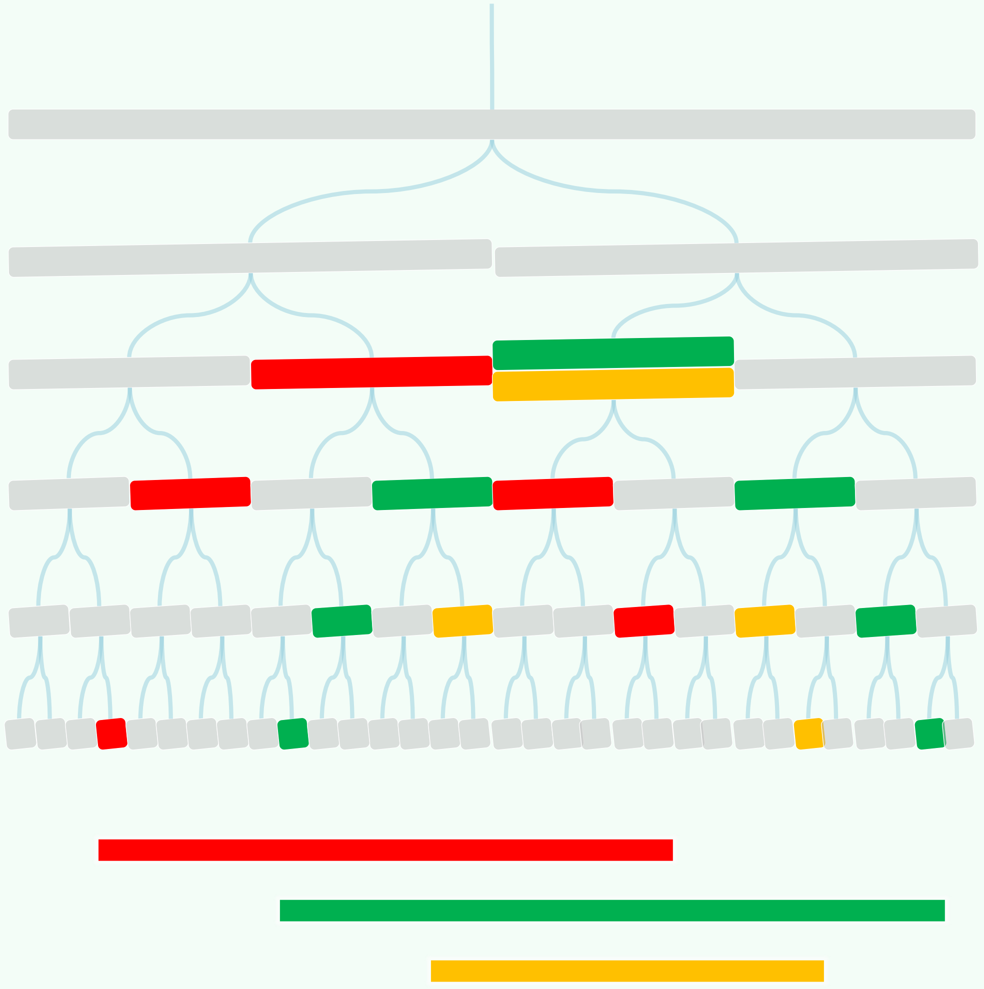
❖ 在树中的每一层，至多访问4个节点

(2个存放s的复本，另2个递归)

故每段区间s的插入，只需$O(\log n)$时间

查询算法：Query(v, q_x)

```
report all the intervals in Int(v)

if ( v is a leaf )

   return

if ( qx ∈ R( lc(v) ) )

   Query( lc(v), qx )

else //qx ∈ R( rc(v) )

   Query( rc(v), qx )
```



查询时间：$\mathcal{O}(r + \log n)$

❖ 树中的每层仅需访问一个节点

累计$\mathcal{O}(\log n)$个

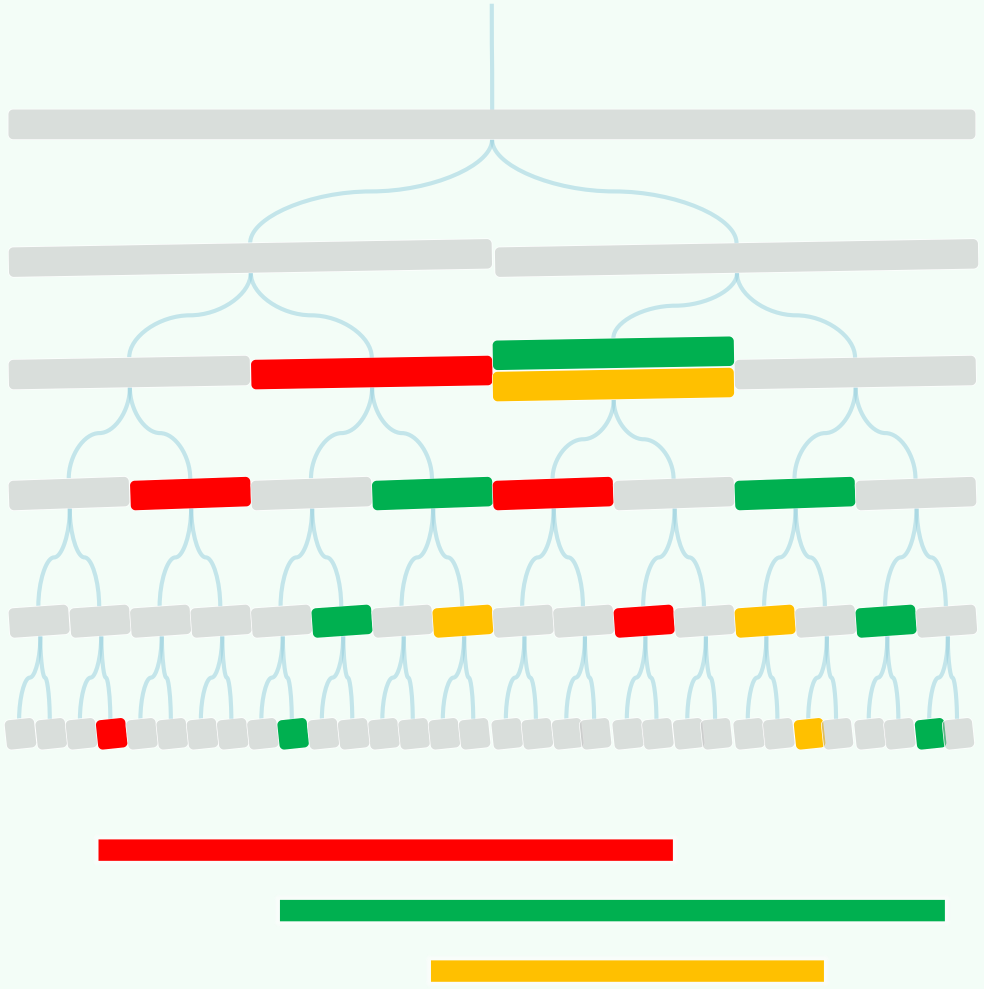
❖ 在每个节点v处

为输出Int(v)所需的时间仅为

$$1 + |\mathrm{Int}(v)| = \mathcal{O}(1 + r_v)$$

❖ 因此，为输出所有命中区间

仅需 $\mathcal{O}(r)$ 时间



结论

❖ 任给x轴上的n段区间

- 可以在$\mathcal{O}$(nlogn)时间内构造出

- 一棵占用$\mathcal{O}$(nlogn)空间的线段树

- 借助它，每次穿刺查询都能

  在$\mathcal{O}$(r + logn)时间内完成