

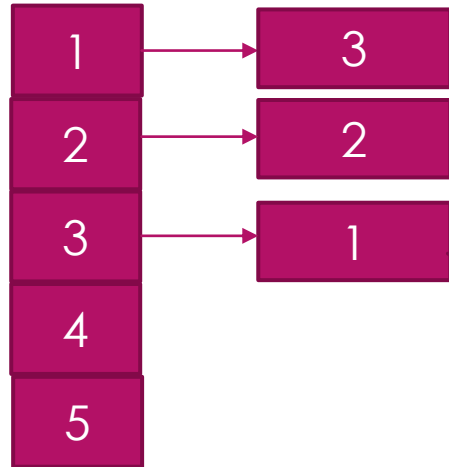
Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8

for a node  $(l,r)$ , which queries can be updated, how to update their answers:

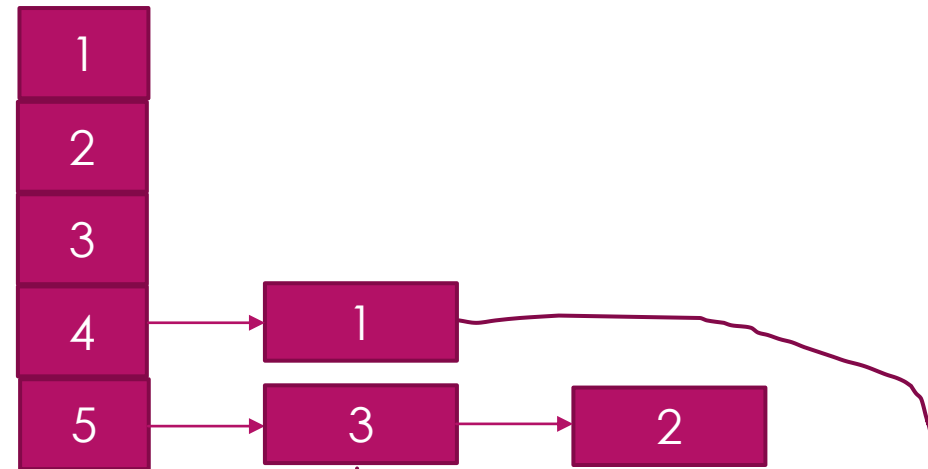
1.  $l < ql \leq \text{mid} < r \leq qr$  : try use right\_result, across[ql,r] to update ;
2.  $ql \leq l \leq \text{mid} < qr < r$  : try use left\_result, across[l,qr] to update;
3.  $l < ql \leq \text{mid} < qr < r$ : try use across[ql,qr] to update.

how to update efficiently

left key



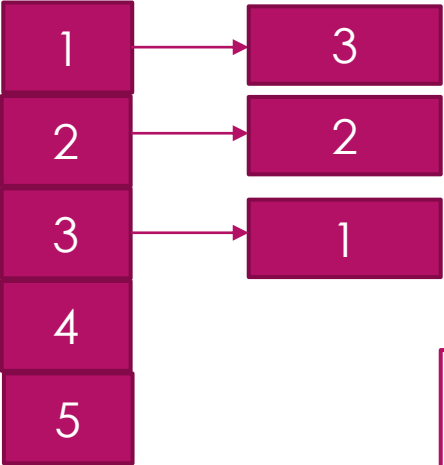
right key



Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8



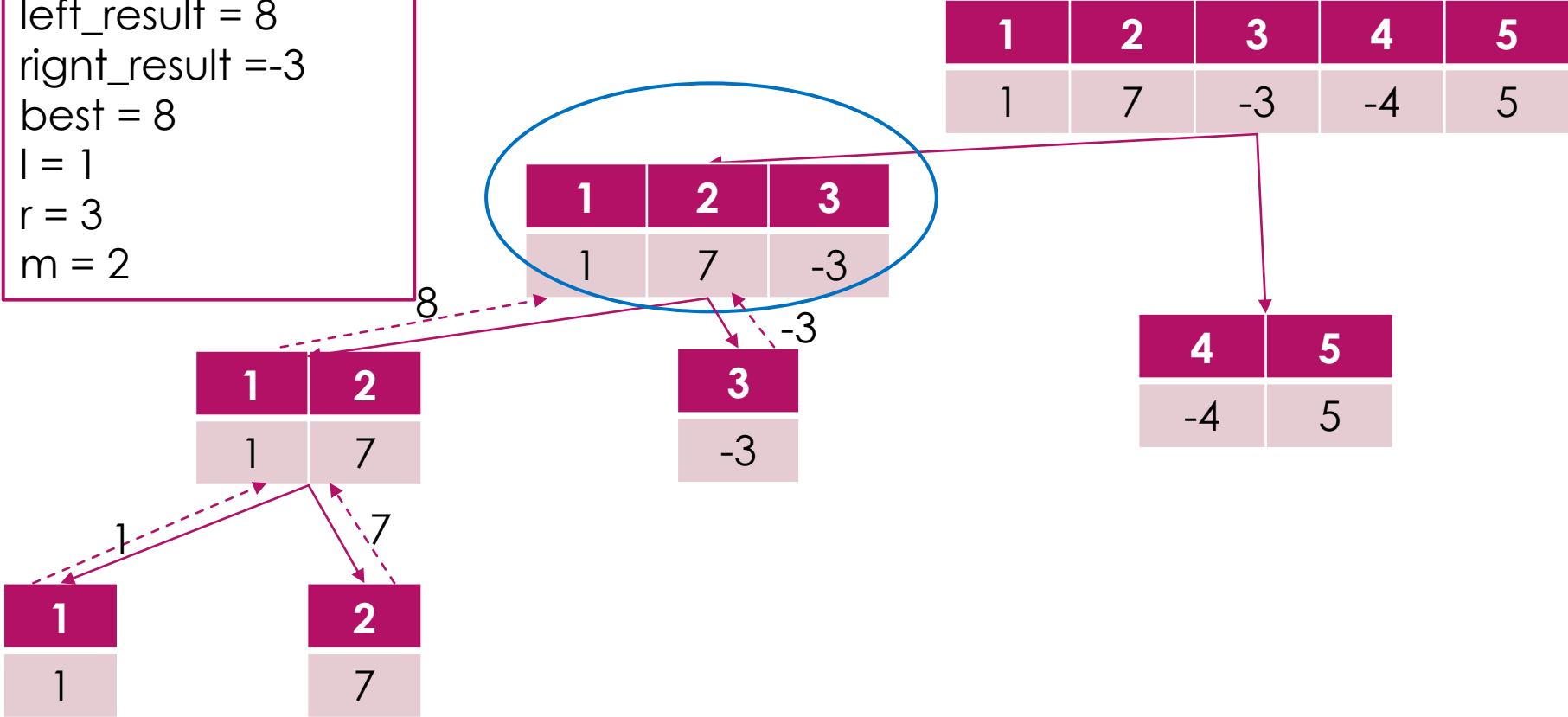
left key



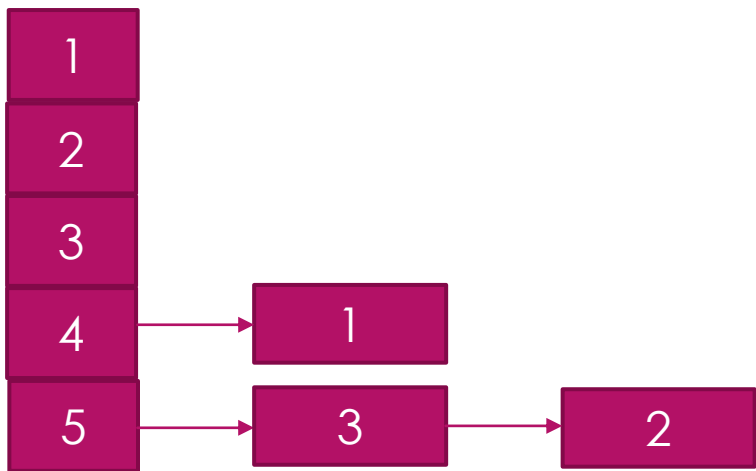
从l到m，  
key=1, 从上述结构  
找到满足的区间定位3, 找到3  
号区间  
(1,5)发现这个区间的qr>r, 用  
right\_result, across[ql,r] 更新  
其answer;  
key= 2,从上述结构找到满足  
的区间定位2, 找到2号区间  
(2,5)发现这个区间的qr>r,用  
right\_result, across[ql,r] 更新  
其answer。

max\_cross\_mid = 5  
left\_result = 8  
right\_result = -3  
best = 8  
l = 1  
r = 3  
m = 2

Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8



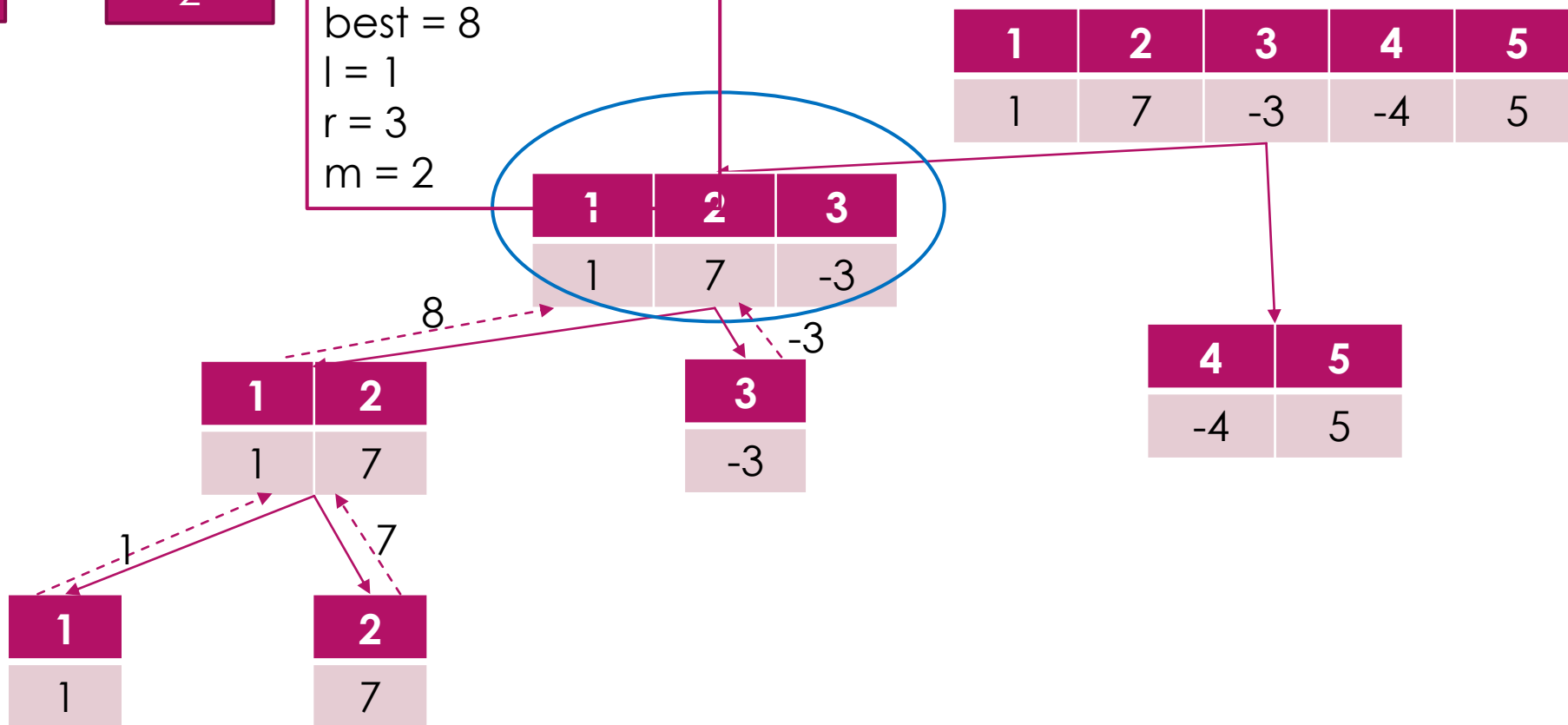
right key



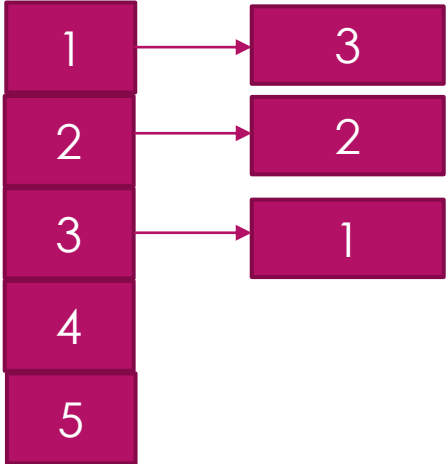
从mid+1到r,  
key=3, 从上述结构  
找不到满足的区间, 不用更新

Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8

max\_cross\_mid = 5  
left\_result = 8  
right\_result = -3  
best = 8  
l = 1  
r = 3  
m = 2



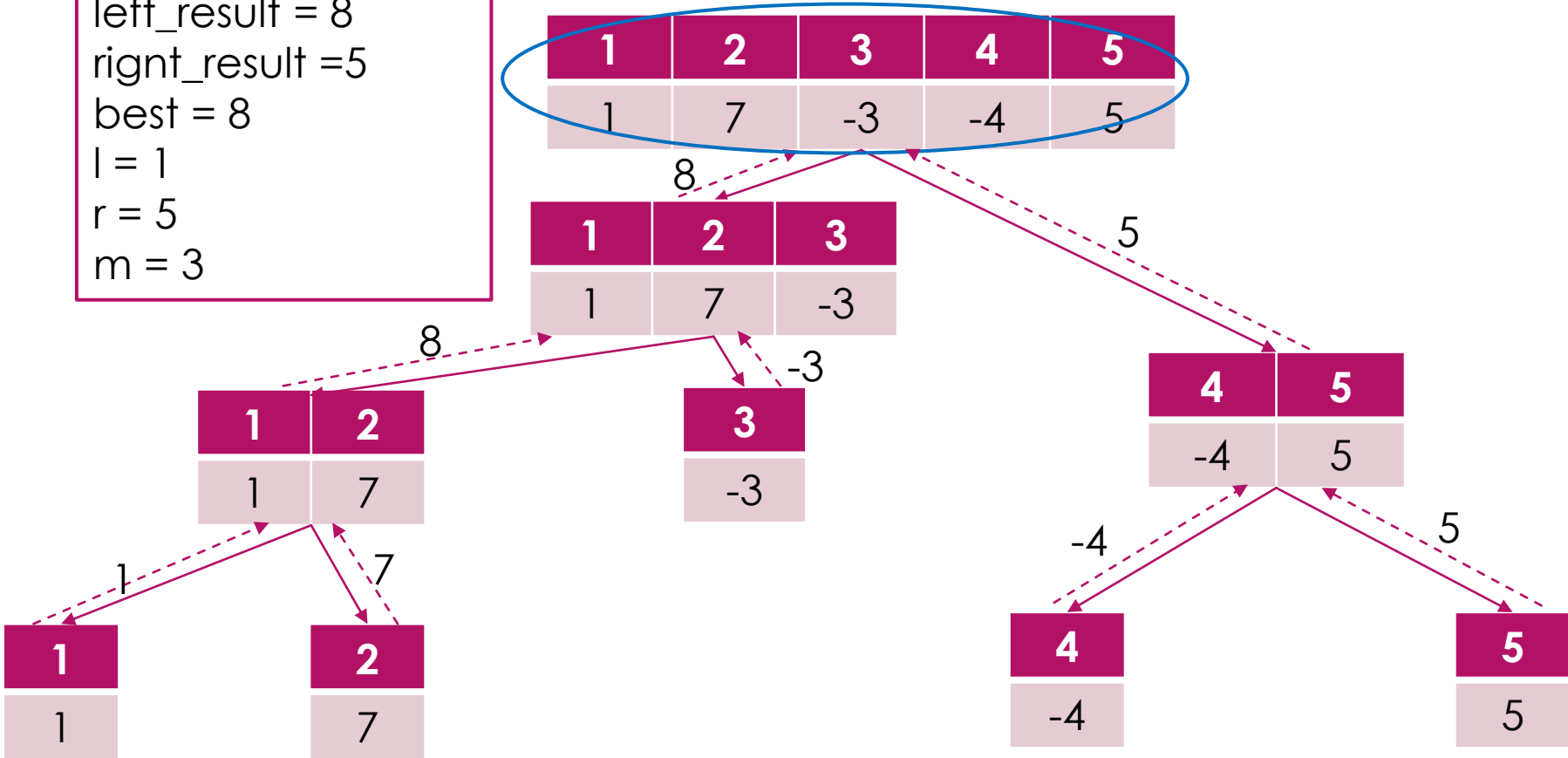
left key



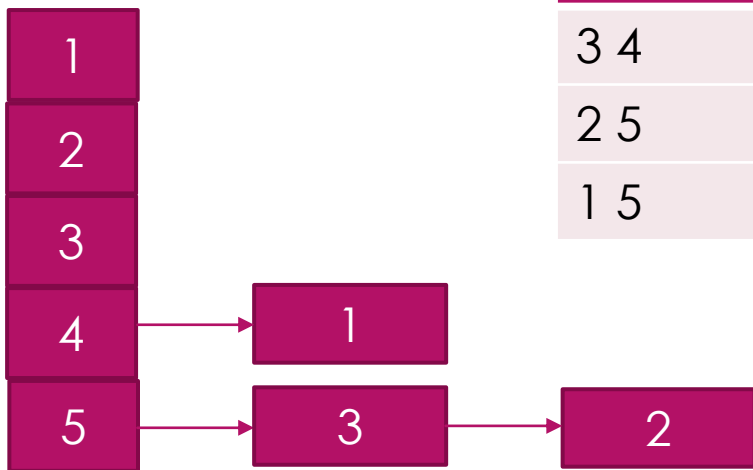
Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8

从l到m，  
key=1, 从上述结构  
找到满足的区间定位3, 找到3  
号区间  
(1,5)发现这个区间的 $qr \geq r$ , 用  
right\_result, across[ql,r] 更新  
其answer;  
key= 2,从上述结构找到满足  
的区间定位2, 找到2号区间  
(2,5)发现这个区间的 $qr \geq r$ ,用  
right\_result, across[ql,r] 更新  
其answer。  
key= 3,从上述结构找到满足  
的区间定位1, 找到1号区间  
(3,4)发现这个区间的 $qr < r$ ,用  
across[ql,qr] 更新其answer。

max\_cross\_mid = 6  
left\_result = 8  
right\_result = 5  
best = 8  
l = 1  
r = 5  
m = 3

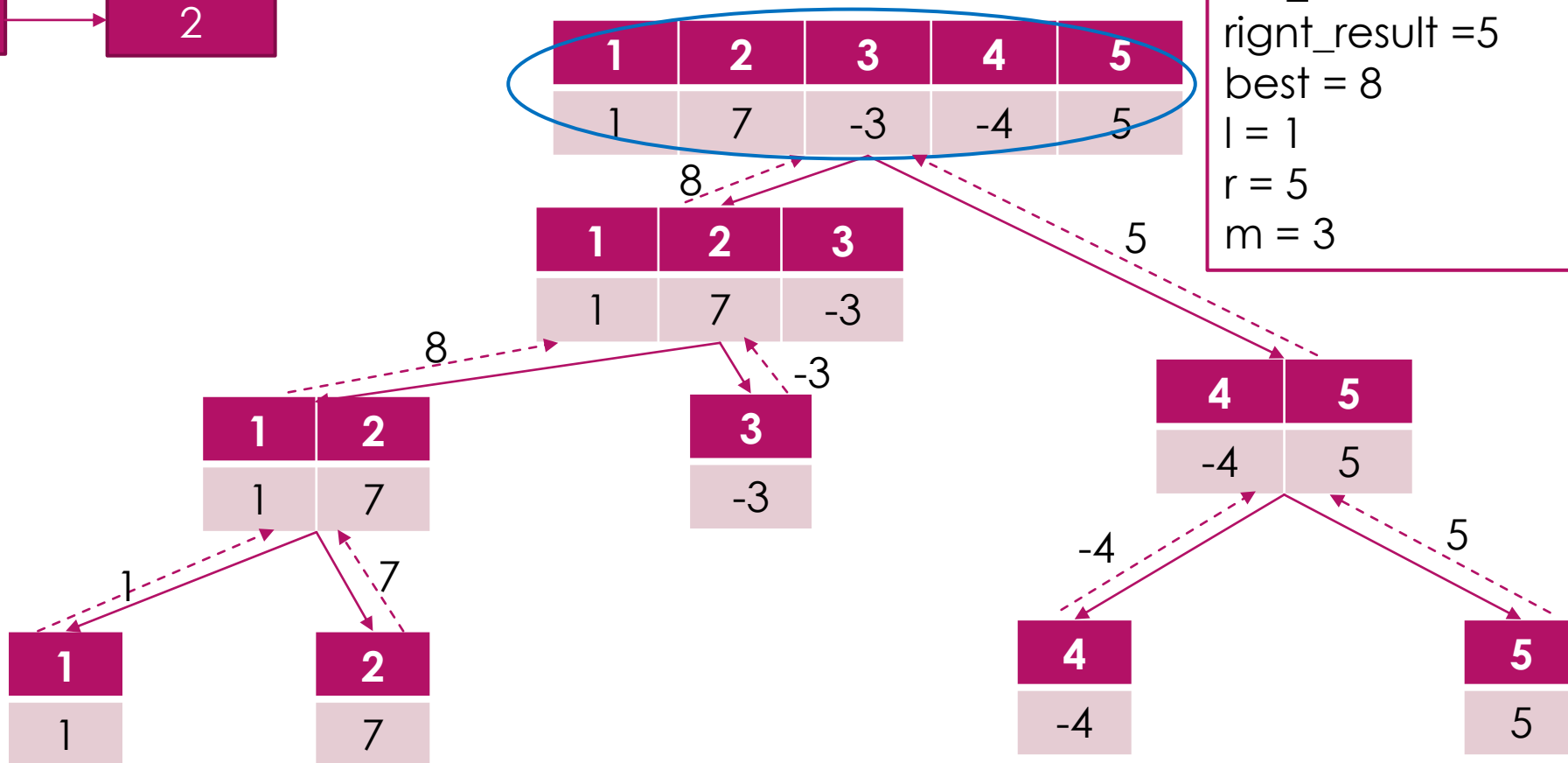


right key



从mid+1到r,  
key=4, 从上述结构  
找到满足的区间定位1, 找1号  
区间  
(3,4)发现这个区间的 $l < ql$ , 用  
across[ql,qr] 更新其answer;  
key= 5,从上述结构找到满足  
的区间定位2和3, 首先找到3  
号区间(1,5)发现这个区间的  
 $ql \leq l$ ,用left\_result, across[l,qr]  
更新其answer;找到2号区间  
(2,5)发现这个区间的 $ql > l$ ,用  
across[ql,qr] 更新其answer。

Query	ql	qr	answer
3 4	3	4	-3
2 5	2	5	7
1 5	1	5	8



max\_cross\_mid = 6  
left\_result = 8  
right\_result = 5  
best = 8  
l = 1  
r = 5  
m = 3