

Range Trees

IOI Training Camp 2
2006

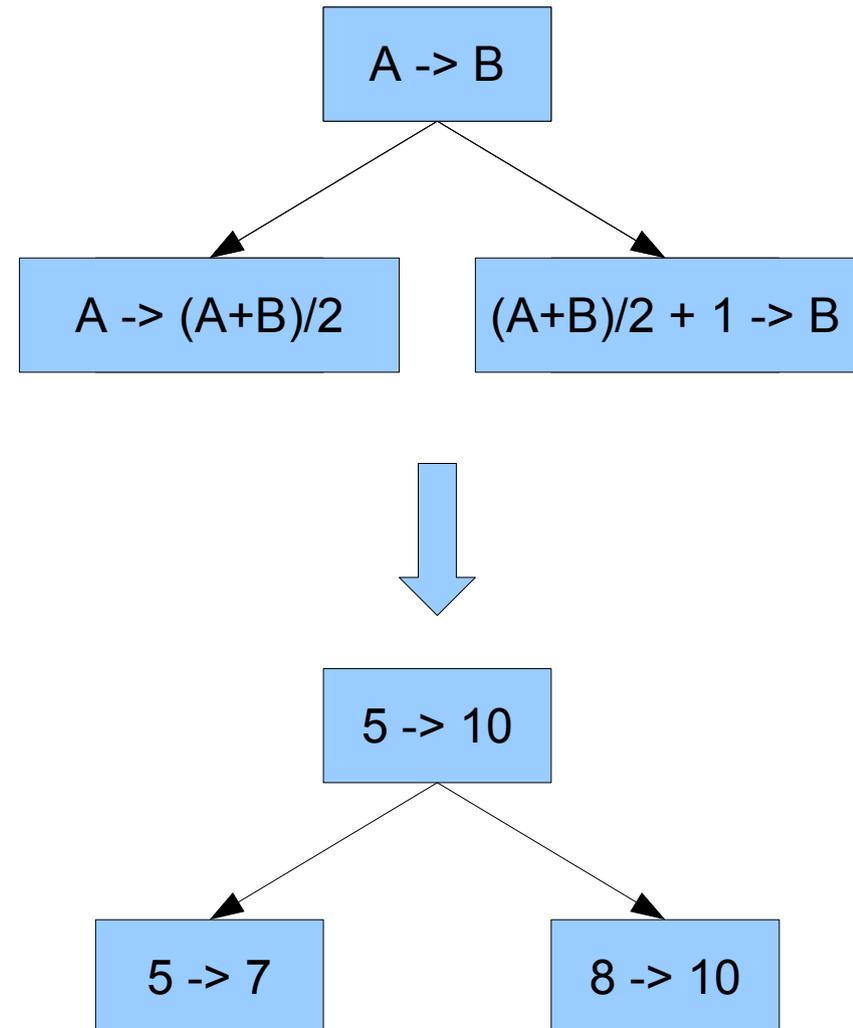
Keegan C-Smith

Overview

- Not called a range tree.
- Range trees can tell you how many of an item you in $O(\log n)$
- Insertions in avg case: $O(\log n)$
- Insertions can be ranges or single number.
- All ranges must be Integer ranges.

How it works

- Complete binary tree.
- Root encompasses entire range.
- Each node will cover a range $[A, B]$.
 - Left child will have range $[A, (A+B)/2]$.
 - Right child will have range $[(A+B)/2, B]$.
 - If $A == B$, a node has no children.



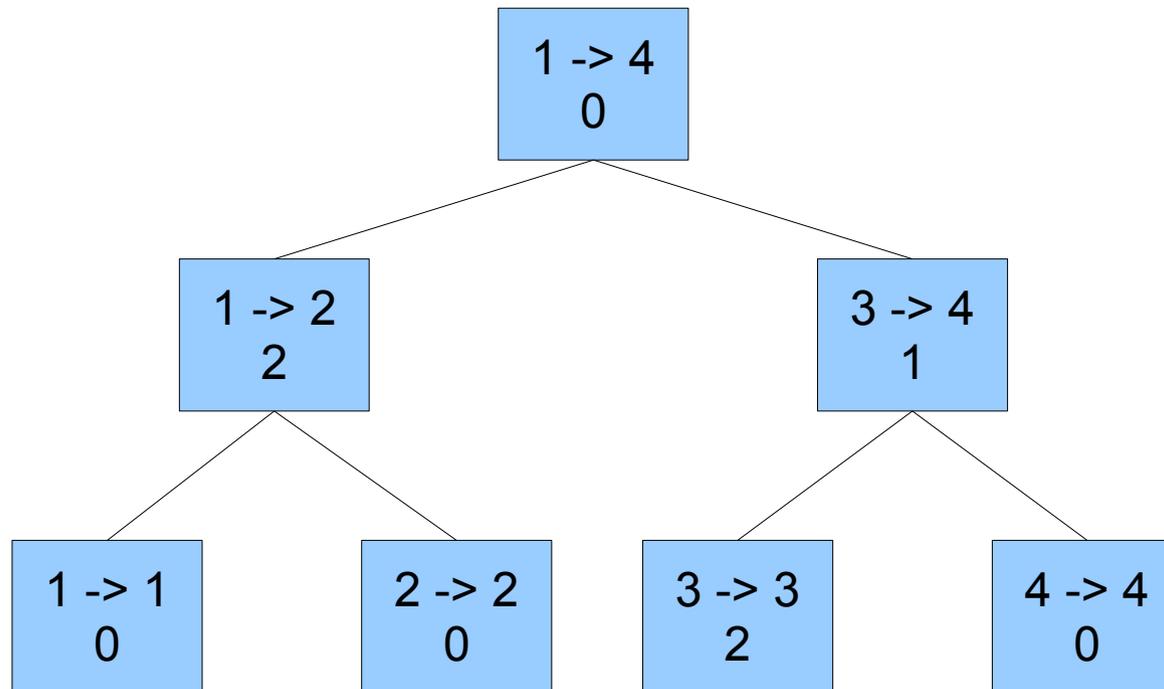
How it works cont.

- Because ranges have to be indexes, $(A+B)/2$ must be floored or ceil. For these slides I am going to use floor.
- Ranges must be inclusive.

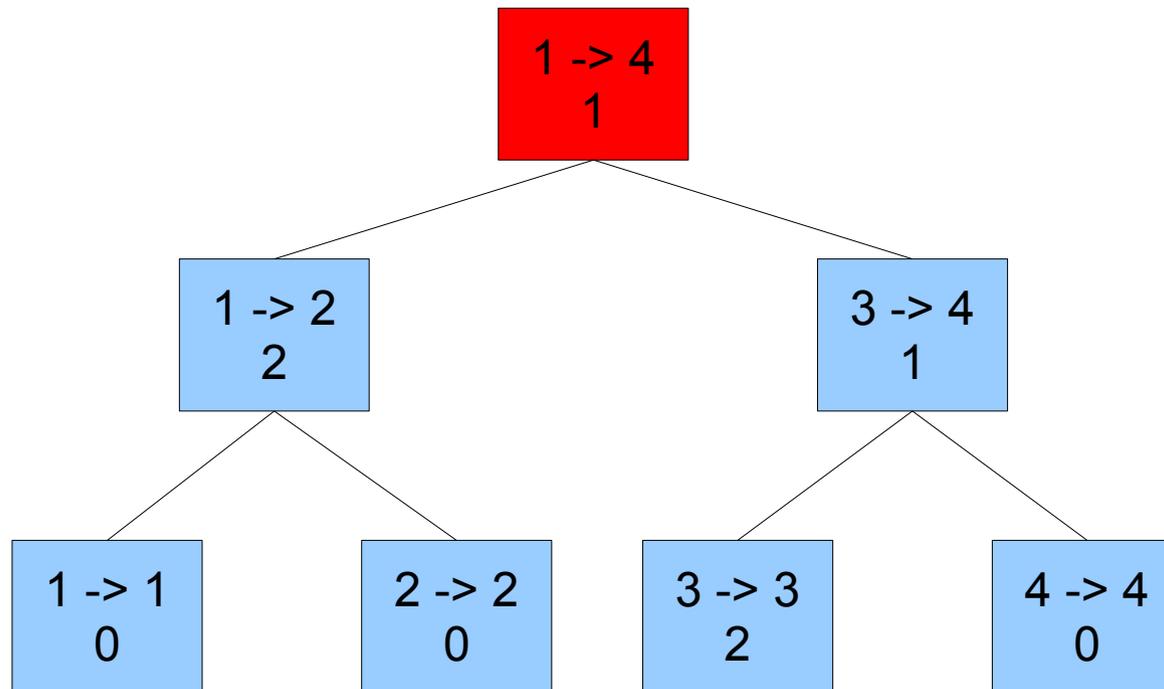
Inserting

- When passing through a node:
 - If every num in range is in nodes range, increase node's count. Do not traverse children.
 - If no num in range is in left child's range, do not traverse left child.
 - If no num in range is in right child's range, do not traverse right child.

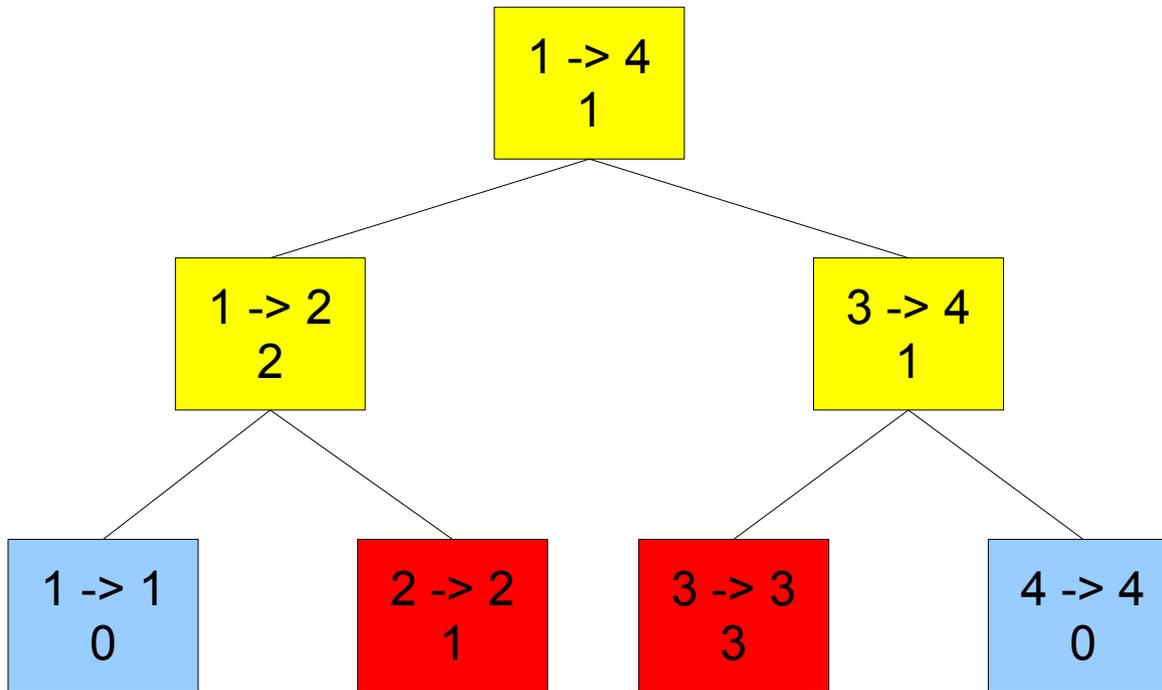
Inserting examples



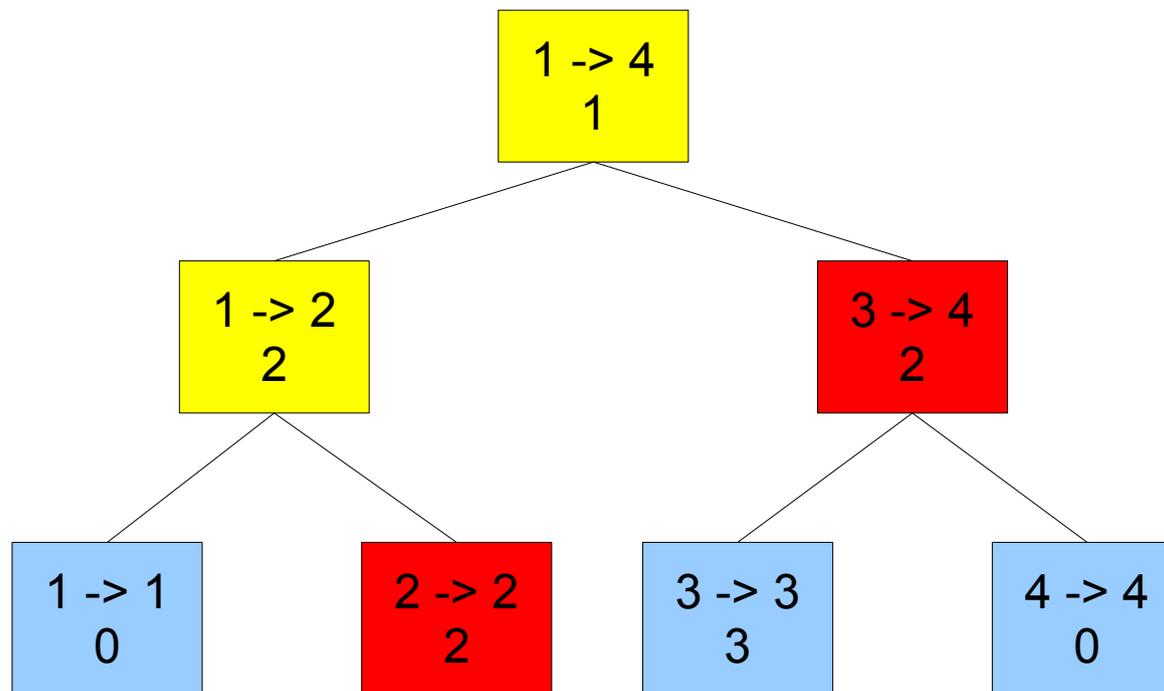
Inserting 1->4



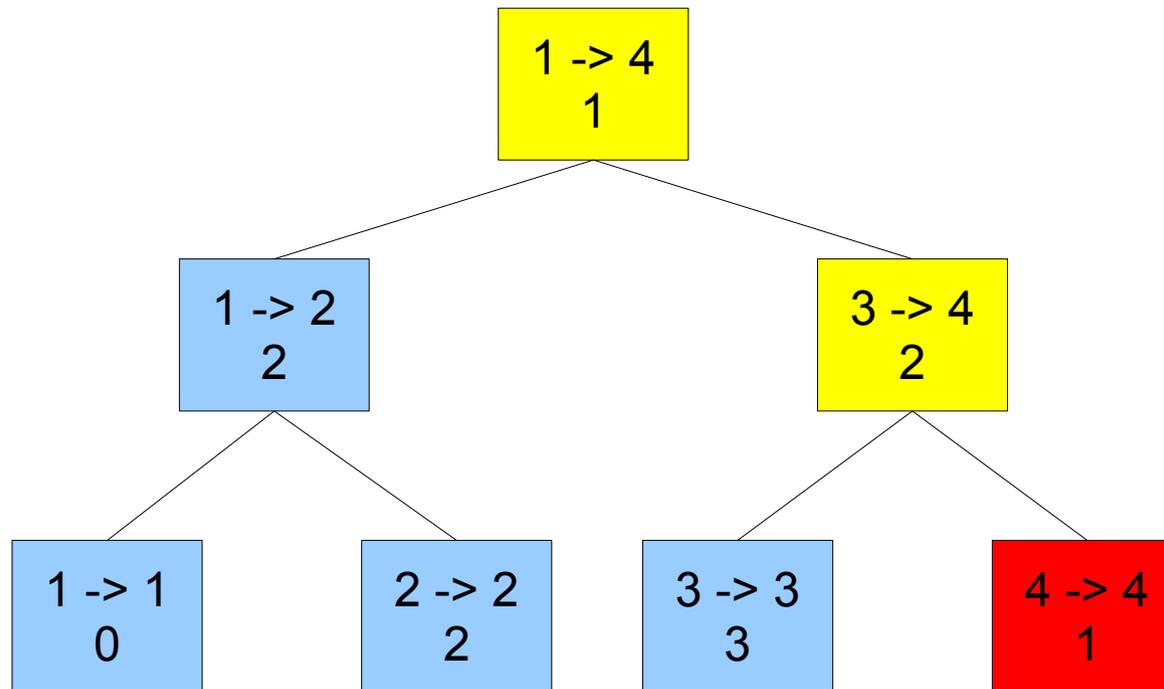
Inserting 2->3



Inserting 2->4



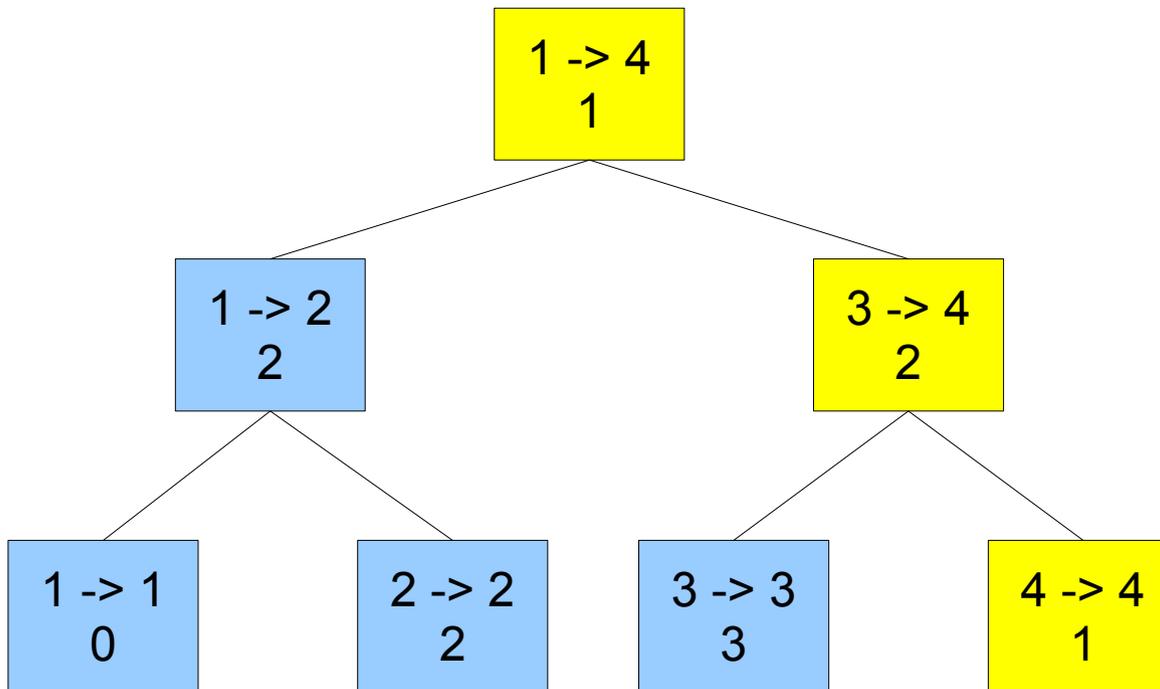
Inserting 4->4



Reading

- To find out how many of item C you have you traverse as if inserting range C->C.
- Each node you pass through, you increment a count.

Reading of 4



There are $1 + 2 + 1 = 4$ items of item 4

Implementation in an Array

- Can be implemented in an array if using indexing from 1.
 - Left tree index = current index * 2
 - Right tree index = current index * 2 + 1
- Size of array: $\sum_{i=0}^{\text{ceil}(\log_2 n)} 2^i$

Array Indexes

