More on Binary Search Trees

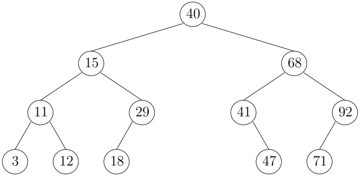
CSCI2100 Tutorial 10

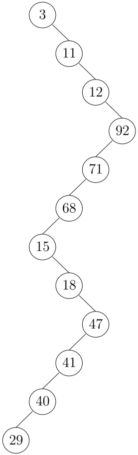
Shangqi Lu

Department of Computer Science and Engineering
The Chinese University of Hong Kong

Binary Search Tree Example

Two possible BSTs on $S = \{3, 11, 12, 15, 18, 29, 40, 41, 47, 68, 71, 92\}$:



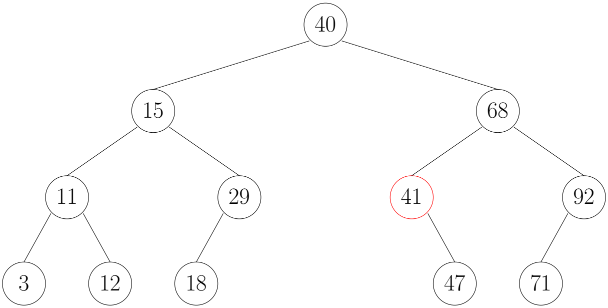


Predecessor Query

Let S be a set of integers. A predecessor query for a given integer q is to find its predecessor in S, which is the largest integer in S that does not exceed q.

Example

Suppose that $S = \{3, 11, 12, 15, 18, 29, 40, 41, 47, 68, 71, 92\}$ and we have a balanced BST T on S:



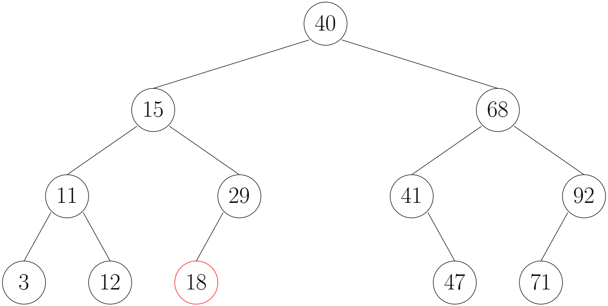
We want to find the predecessor of $q = 42$ in S.

Successor Query

Let S be a set of integers. A successor query for a given integer q is to find its successor in S, which is the smallest integer in S that is no smaller than q.

Example

We want to find the successor of $q = 17$ in S.



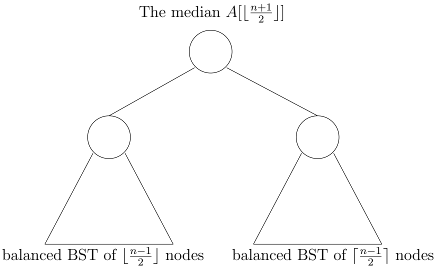
Construction of a Balanced BST

In the following, we will discuss how to construct a balanced BST T on a given sorted set S of n integers in $O(n)$ time.

Construction of a Balanced BST

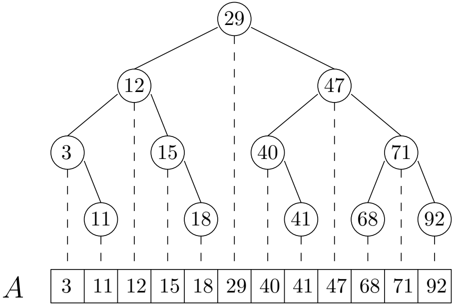
Assume that S is stored an array A and A is sorted.

- **Observation:** The subtree of any node in a balanced BST is also a balanced BST.
- **Main idea:** A BST of n nodes constructed by the following form:



Example

Let us construct a balanced BST T on the following sorted array A.



Construction of a Balanced BST

Let $f(n)$ be the maximum running time for constructing a balanced BST from an array of length n. We have:

$$f(1) = O(1)$$
$$f(n) = O(1) + 2 \cdot f(\lceil n/2 \rceil)$$

Solving the recurrence gives $f(n) = O(n)$.

Range Count Problem

Let S be a set of n integers. Given two integers a and b such that $a \leq b$. Find the number of integers in S which are in the range of $[a, b]$.

In the following, we will discuss how to augment a balanced BST on S to achieve:

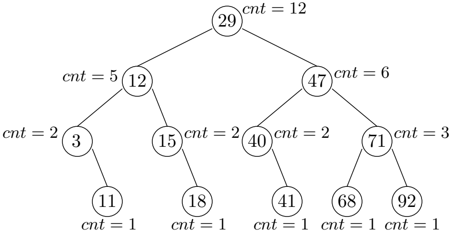
- $O(n)$ space consumption,
- $O(\log n)$ time for each query.

Range Count Problem

Augment a balanced BST T on S by storing one additional information in each node u that is:

- the number of nodes in the subtree of u.

For example,

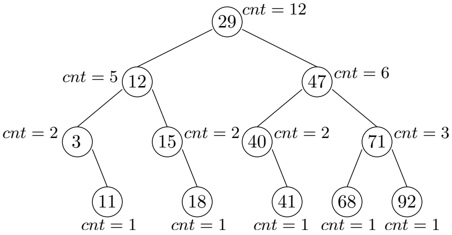


Range Count Problem

Define a concept first.

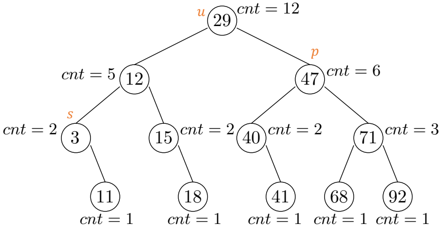
- Lowest Common Ancestor: Let t be the root. The lowest common ancestor of nodes v_1 and v_2 is the lowest node that is on both of the paths $P(t, v_1)$ and $P(t, v_2)$.

For example, the lowest common ancestor of node with key 3 and node with key 15 is the node with key 12.



Range Count Problem

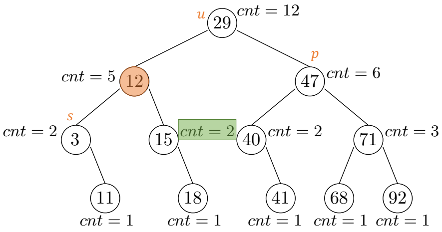
For a range $[2, 48]$, let s be the successor of 2, p the predecessor of 48 and u the lowest common ancestor of s and p.
Initialize a count $c = 1$ (since u is within the range)



Range Count Problem

Traverse the path from u's left child to s.
For every node v being visited, if v.key ≥ 2:

- c $+= 1$
- c $+=$ the counter of v's right child

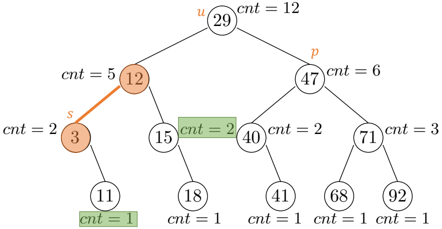


C is incremented by $1 + 2$.

Range Count Problem

Traverse the path from u's left child to s.
For every node v being visited, if v.key ≥ 2:

- c $+= 1$
- c $+=$ the counter of v's right child

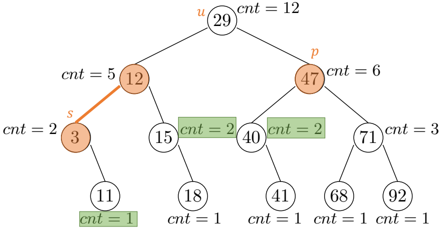


C is incremented by $1 + 1$.

Range Count Problem

Traverse the path from u's right child to p.
For every node v being visited, if v.key ≤ 48:

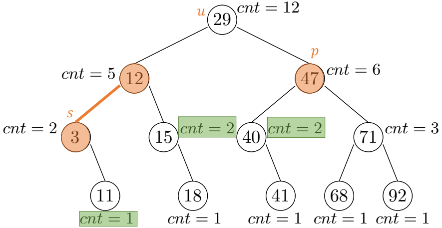
- c $+= 1$
- c $+=$ the counter of v's left child



C is incremented by $1 + 2$. Finally, c becomes 9.

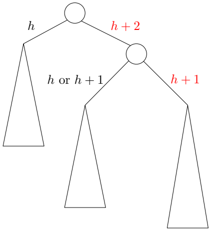
Range Count Problem

We walked through two paths, at most $\log_2 n$ nodes in each path.
For each node visited, we perform constant-time operations, which takes $O(1)$.
Time complexity: $O(\log n)$



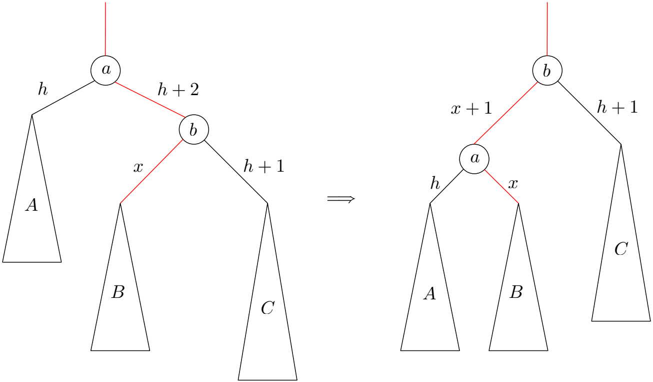
Rebalancing

In lectures we explored the Left-Left and Left-Right cases in detail, so here we will look at Right-Right and Right-Left:



Right-Right

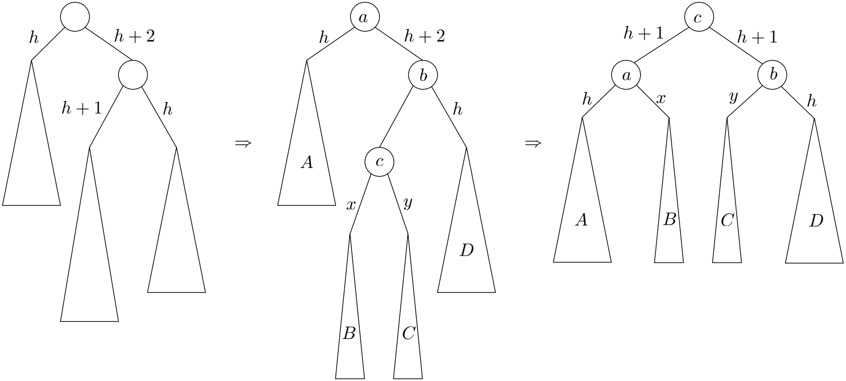
Fix by a rotation (symmetric to left-left):



Note that $x = h$ or $h + 1$, and the ordering from left to right of A, a, B, b, C is preserved after rotation.

Right-Left

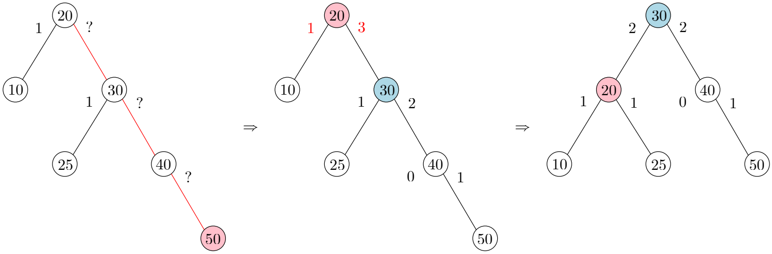
Fix by a double rotation (symmetric to left-right):



Note that x and y must be h or $h-1$. Futhermore at least one of them must be h.

Right-Right Example

Inserting 50:



Right-Left Example

Inserting 38: