Augmented AVL trees: Interval Trees

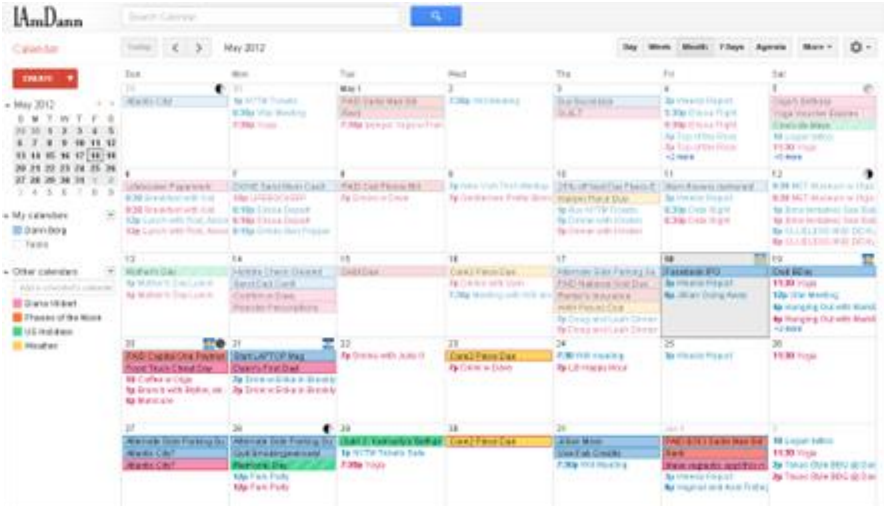
Slides adapted from Trevor Brown

Augmented Trees

- In addition to the value, the node stores additional information
    - Height
        - Saw this in our implementation of AVL trees
    - Average of key/values in subtree rooted in node
    - Maximum of key/value of subtree rooted in node
- Need to maintain the additional information as nodes are added/deleted
- In an AVL tree, need to still perform operations in O(log n) time

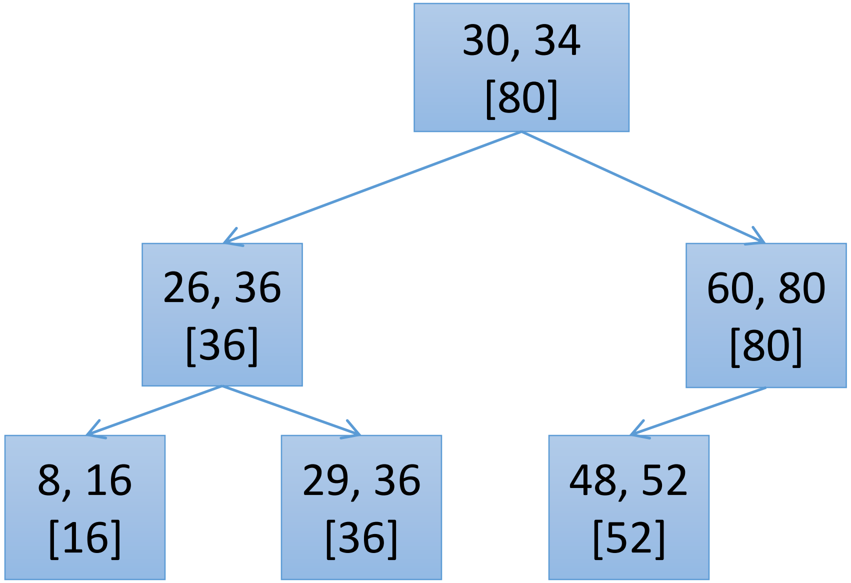
Scheduling conflict problem

- A calendar contain time intervals [lo, hi]
- Want to quickly know whether a new interval conflicts with any existing intervals in the calendar



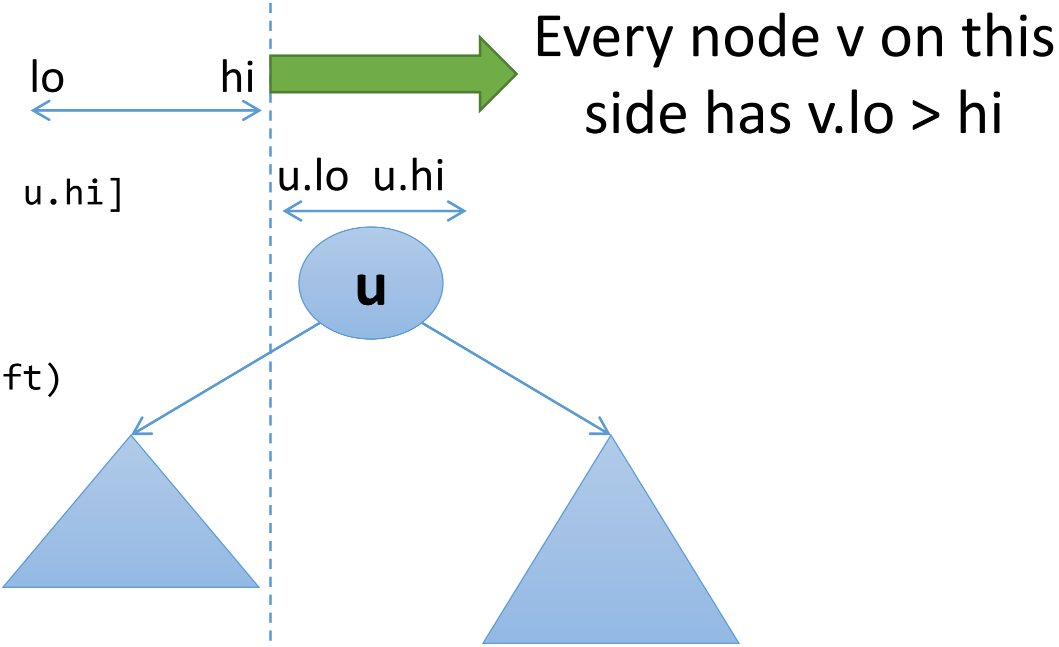
Building the interval tree

- Use the low end-point of the interval as the key
- Store the maximum high end point of any node in the subtree

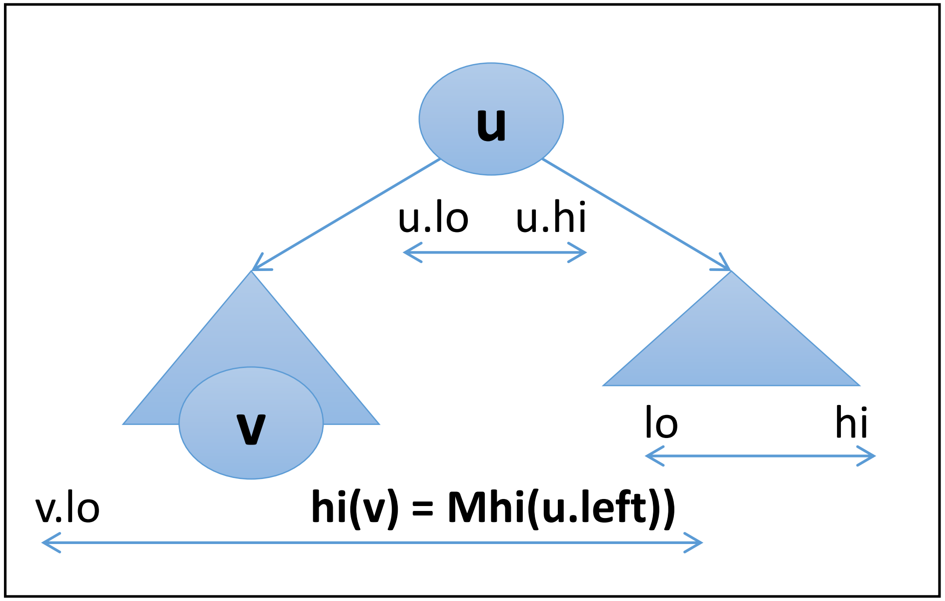


```
Search(lo, hi, u)
  # Return an interval that intersects [lo hi] if
  # it exists in the subtree node, otherwise return
  # null
  if u is null
     return null
  if [lo, hi] intersects [u.lo, u.hi]
     return [u.lo, u.hi]
  if lo < u.lo
     return Search(lo, hi, u.left)
  else
     if lo >  u.left.Mhi
       return Search(lo, hi, u.right)
     else
       return Search(lo, hi, u.left)

Search(lo, hi, root)
Search(lo, hi, u)
  # Return an interval that intersects [lo hi] if
  # it exists in the subtree node, otherwise return
  # null
  if u is null
     return null
  if [lo, hi] intersects [u.lo, u.hi]
     return [u.lo, u.hi]
  if lo < u.lo
     return Search(lo, hi, u.left)
```



```
else
    if lo >  u.left.Mhi
      return Search(lo, hi, u.right)
    else
      return Search(lo, hi, u.left)
```



Insert algorithm

Insert(node, lo, hi)

- Create a new node with key lo, also storing hi
- Set Mhi of the new node to hi
- Insert the node into the AVL tree
- With every rotation, update Mhi using

  Mhi(u) = max(hi(u), Mhi(left(u)), Mhi(right(u))