

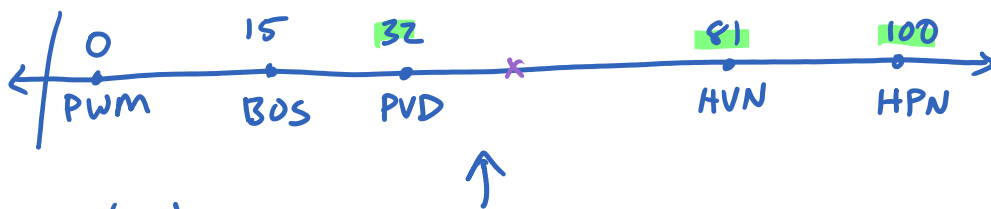
Map Implementation Summary

	unsorted list (array/linked)	sorted list (array)	sorted list (linked list)	hash table
contains/get	$O(n)$	$O(\log n)$	$O(n)$	$O(1)$ expected $O(n)$ worst-case
put	$O(n)$	$O(n)$	$O(n)$	$O(1)$ expected $O(n)$ worst-case
remove	$O(n)$	$O(n)$	$O(n)$	$O(1)$ expected $O(n)$ worst-case
for-each	$O(n)$	$O(n)$	$O(n)$	$O(m+n) = O(n)$
keys-sorted	$O(n \log n)$	$O(n)$	$O(n)$	$O(n \log n)$

if hash fn distributes keys uniformly randomly and $C_1 \leq d \leq C_2$

*for each chain
for each node on chain
 $f(\text{node} \rightarrow \text{key}, \text{node} \rightarrow \text{value})$*

Problem: Given locations of stations on number line, and point x, find closest station to left and/or right and its id.



keys: locations

values: ids of what's at those locations

	key	value
0	0	PWM
1	15	BOS
2	100	HPN
3		
4	32	PVD
5	81	HVN
6		



Problem: Given intervals of memory addresses, userid of owners of those intervals, and memory addr, find the owner of the containing interval.

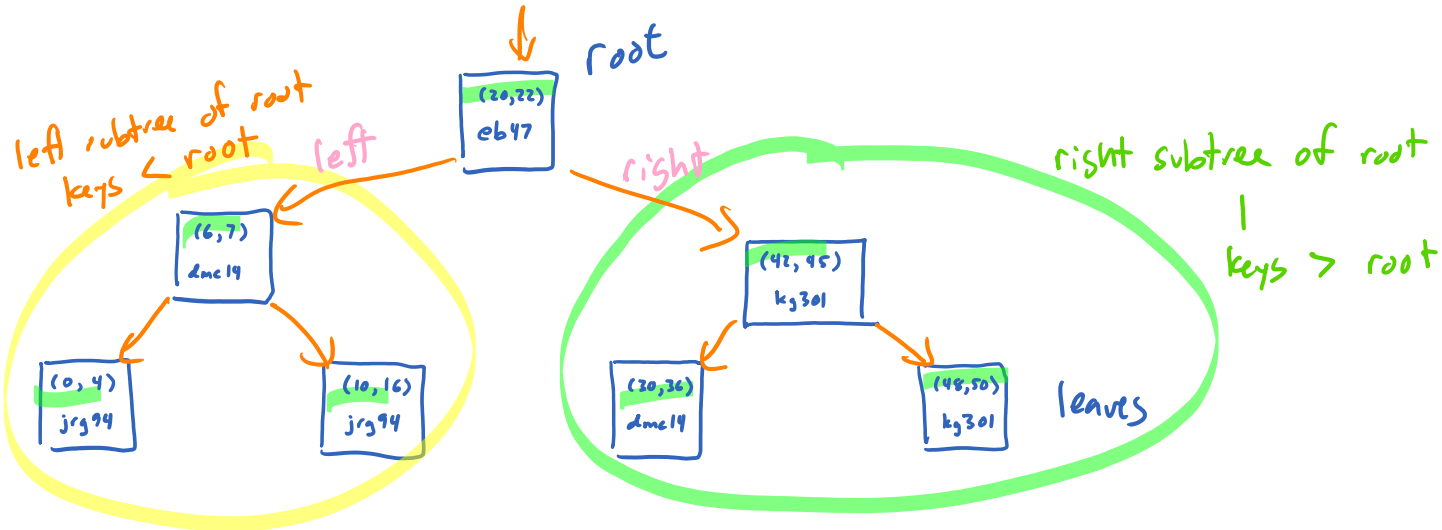
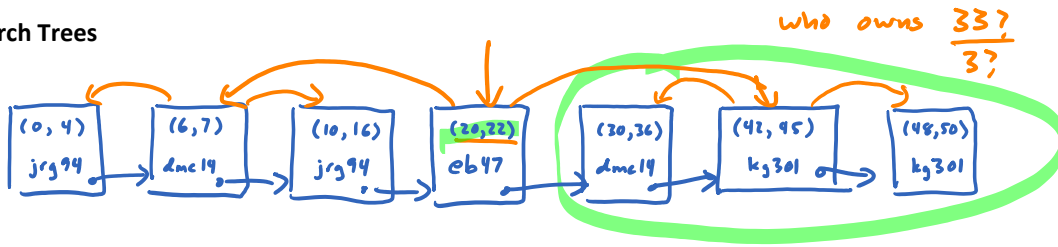
keys: intervals

values: userids

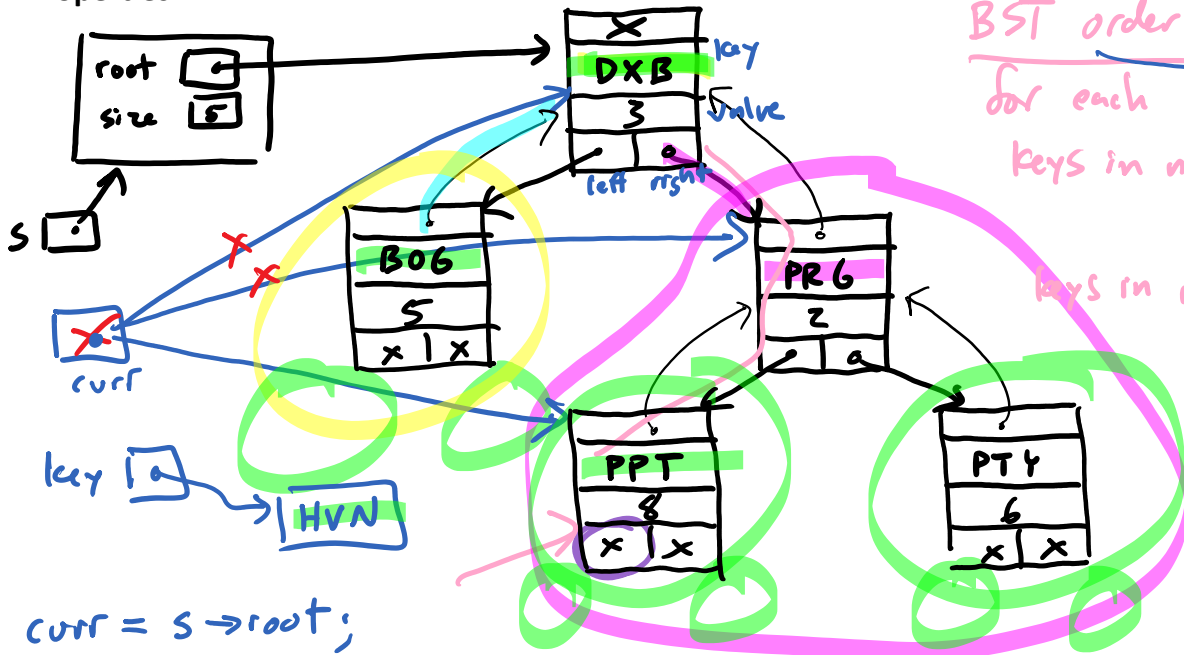
who owns 90?

(10,20)	jq301
(30,60)	cb203
(61,100)	jq301
(105,120)	kg610

Binary Search Trees



BST Properties

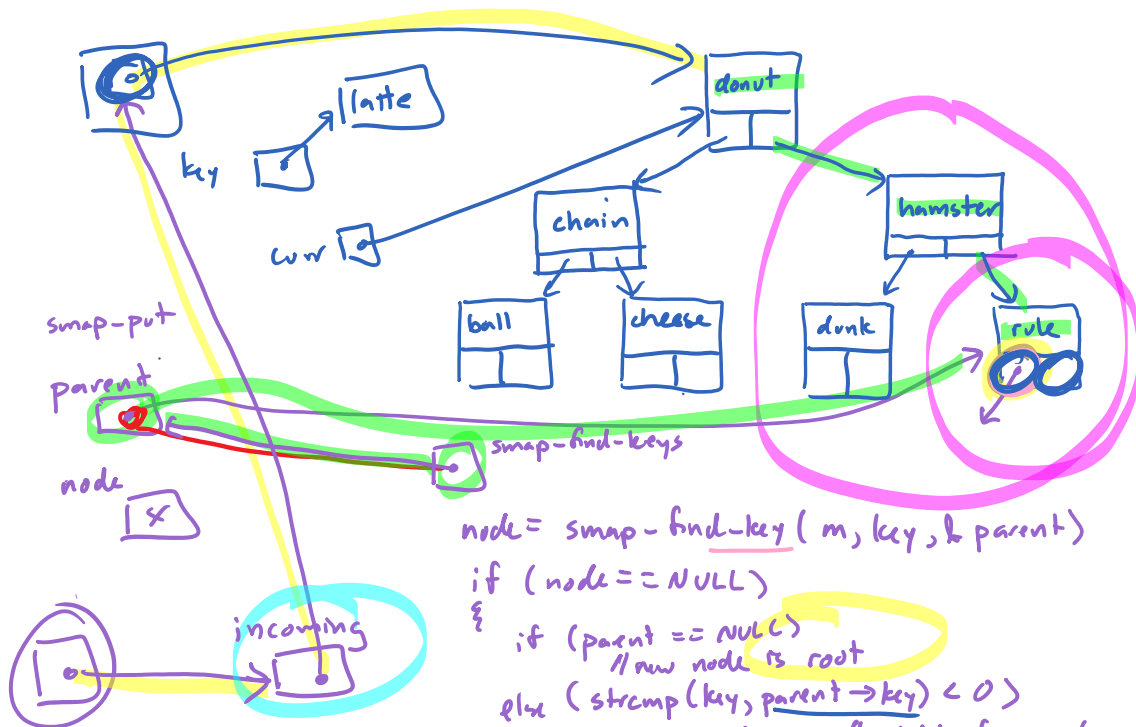


BST order property
 for each node n
 keys in n 's left subtree $<$ n 's key
 keys in n 's right subtree $>$ n 's key

```

curr = s -> root;
while (curr != NULL && strcmp(curr->key, key) != 0)
{
    if (strcmp(curr->key, key) < 0)
        curr = curr->right;
    else
        curr = curr->left;
}
return curr != NULL;
    
```

Adding to a BST



running time
 $O(h)$ $h = \text{height of tree}$

```
node = smap-find-key(m, key, &parent)
```

```
if (node == NULL)
```

```
{ if (parent == NULL)
  // new node is root
```

```
else (strcmp(key, parent->key) < 0)
```

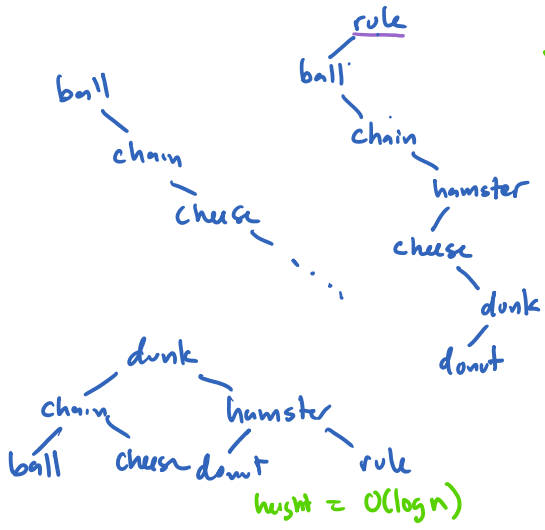
```
// new node is left child of parent
```

```
else
  // new node is right child of parent
```

```
&incoming = new-node;
```

Unshapely Trees

rule ball chain hamster cheese dunk donut



worst-case height
for BST of n keys
= n
so worst-case time
for put/get/contains
remove
 $O(n)$

