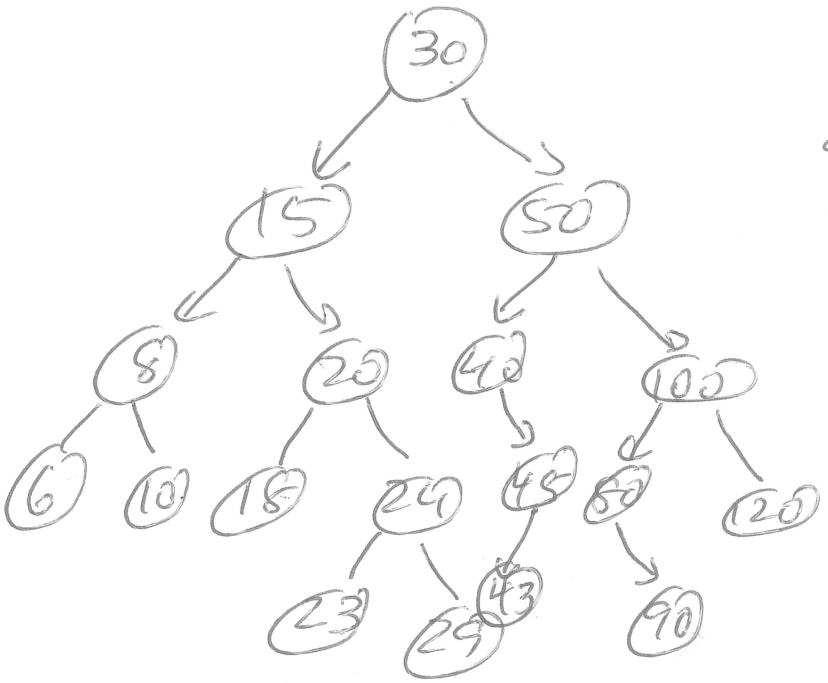
Binary Search Trees

10/24/25 COP3502



BST Node Property - all nodes left subtree less root, all nodes in right subtree are greater than the root

insert (43)

Run-time is $O(h)$

h = height of tree

Insert

```
bintreenode* insert (bintreenode* root, int val) {
    if (root == NULL) return makeNode(val);
    if (val < root->data)
        root->left = insert(root->left, val);
    else
        root->right = insert(root->right, val);
    return root;
}
```

$O(h)$

empty h = -1

h = 0

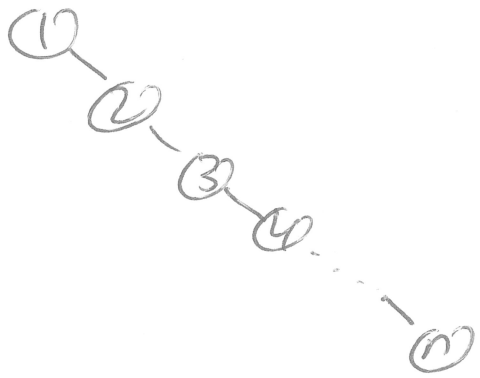
h = 1

height small cases

leaf node - a node w/0 children

height - best + worst case

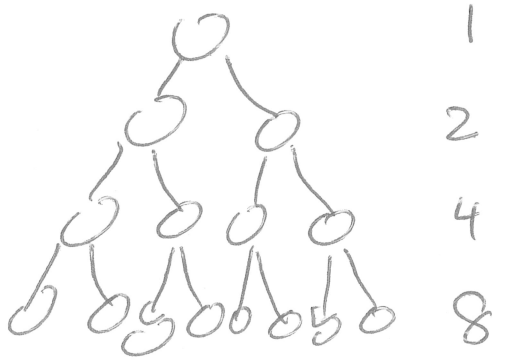
WORST



$h = n - 1$

$= O(n)$

BEST



h ——— 2^h nodes

$n = 1 + 2 + 4 + \cdots 2^h$

$n = 2^{h+1} - 1$

$n+1 = 2^{h+1}$

$h+1 = \log_2(n+1)$

$h = \log_2(n+1) - 1$

$h = O(\lg n)$

Code for height

```
int height (bintreenode* root){
  if (root == NULL)
     return -1;
  int leftH = height(root->left);
  int rightH = height(root->right);
  if (leftH > rightH)
     return leftH +1;
  return rightH+1;
}
```

$O(n)$

```
int search (bintreenode* root, int val) {
    if (root == NULL) return 0;
    if (val < root->data)
        return search (root->left, val);
    else if (val > root->data)
        return search(root->right, val);
    else
        return 1;
}
```

Delete Cases

1) free(delnode);
2) parent->right = NULL;

could be left

CASE 1

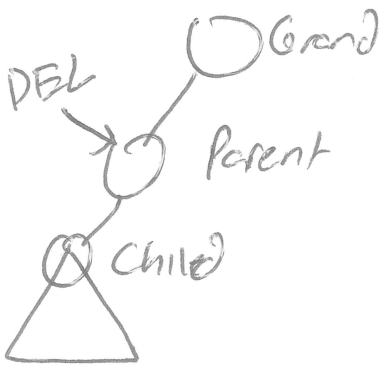
CASE 2 (one child case)

Delete (40)

PATCH PAR OF DEL~~DEL~~NODE TO CHILD OF DELNODE

4 CASES

CASE 2 CASES



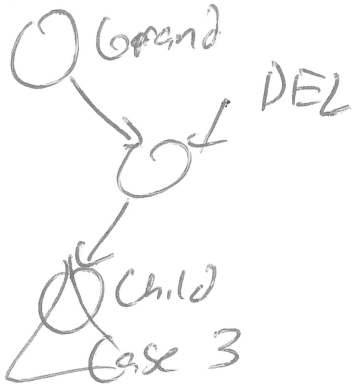
Case 1

grand→left = DEL→left

CASE 2

grand→left =

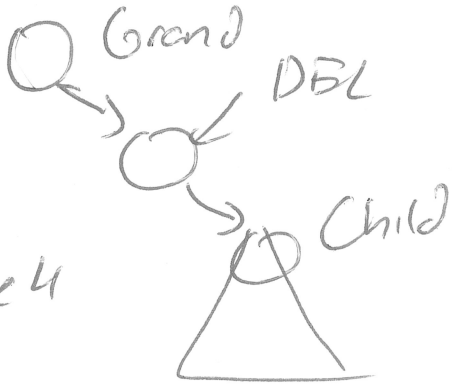
DEL→right



Case 3

grand→right =

DEL→left



Case 4

grand→right = DEL→right;

free (DEL);] in all cases after linking

CASE 3

Candidates to replace 30's position in tree?

→ Ans: Min Val Right

Alt Ans: Greatst on Left

Guaranteed that Min on Right HAS NO left child!

Guaranteed that Max on Left HAS NO right child!

→ both of these are either case 1 or case 2

1) Delete Min on Right OR Max on Left but save value in tmp variable.

2) Copy over saved data into slot for deleted node.