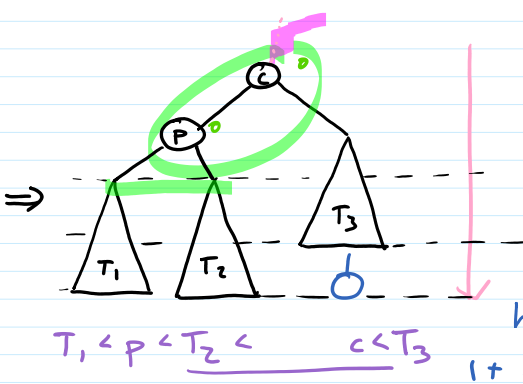
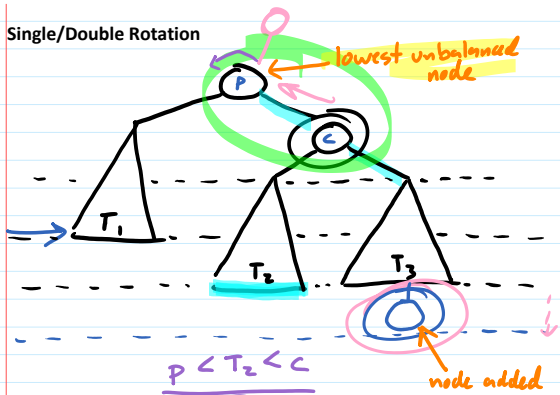
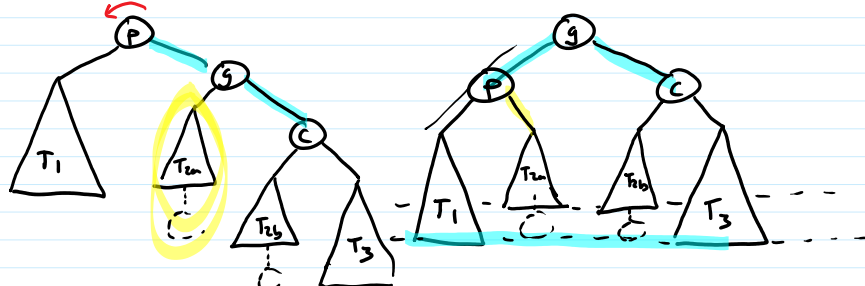
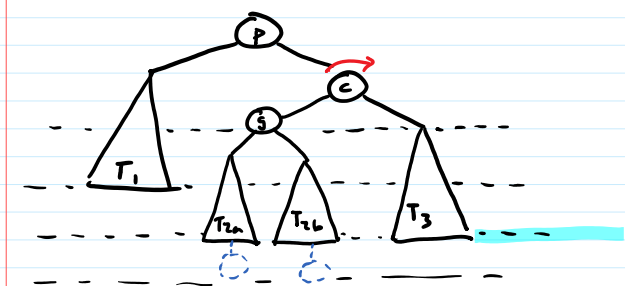
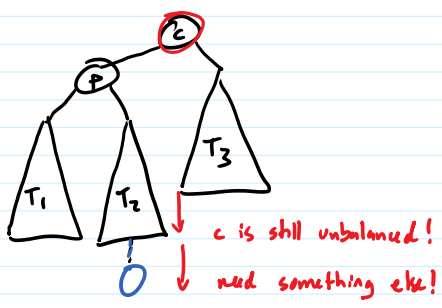
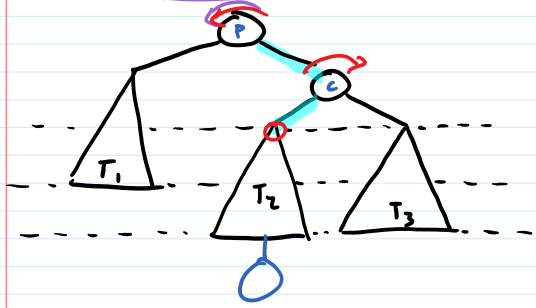


Single/Double Rotation



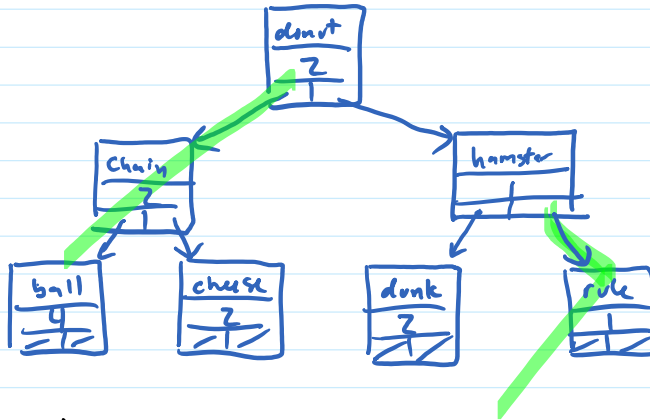
height at p =
 $1 + \max(\text{height at } p \rightarrow \text{left}, \text{height at } p \rightarrow \text{right})$



inorder: $T_1 \ p \ T_{2a} \ g \ T_{2b} \ c \ T_3$

$T_1 \ p \ T_{2a} \ g \ T_{2b} \ c \ T_3$

Inorder Traversal



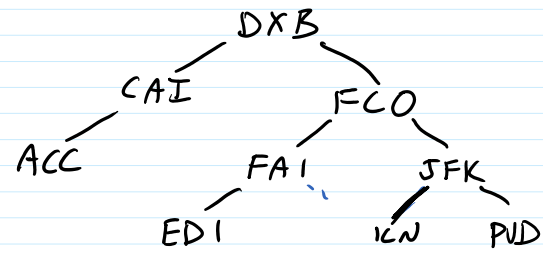
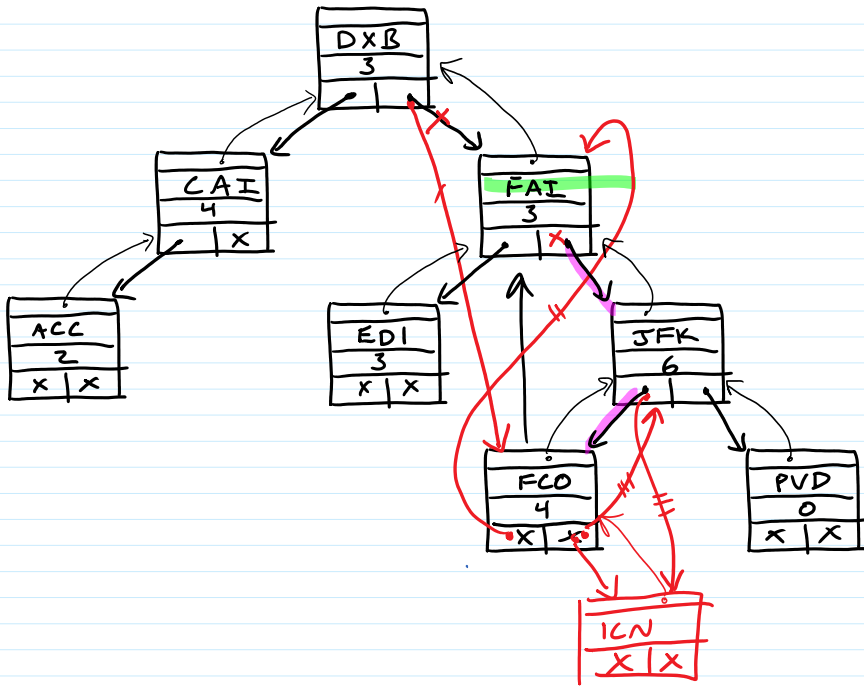
inorder traversal
processes keys in sorted order

- ball
- chain
- cheese
- donut
- dunk
- hamster
- rule

traverse (n)

```
if (n != NULL) {  
    printf("%s\n", n->key)    preorder  
    traverse(n->left)  
    printf("%s\n", n->key)    inorder  
    > traverse(n->right)  
    printf("%s\n", n->key)    postorder  
}
```

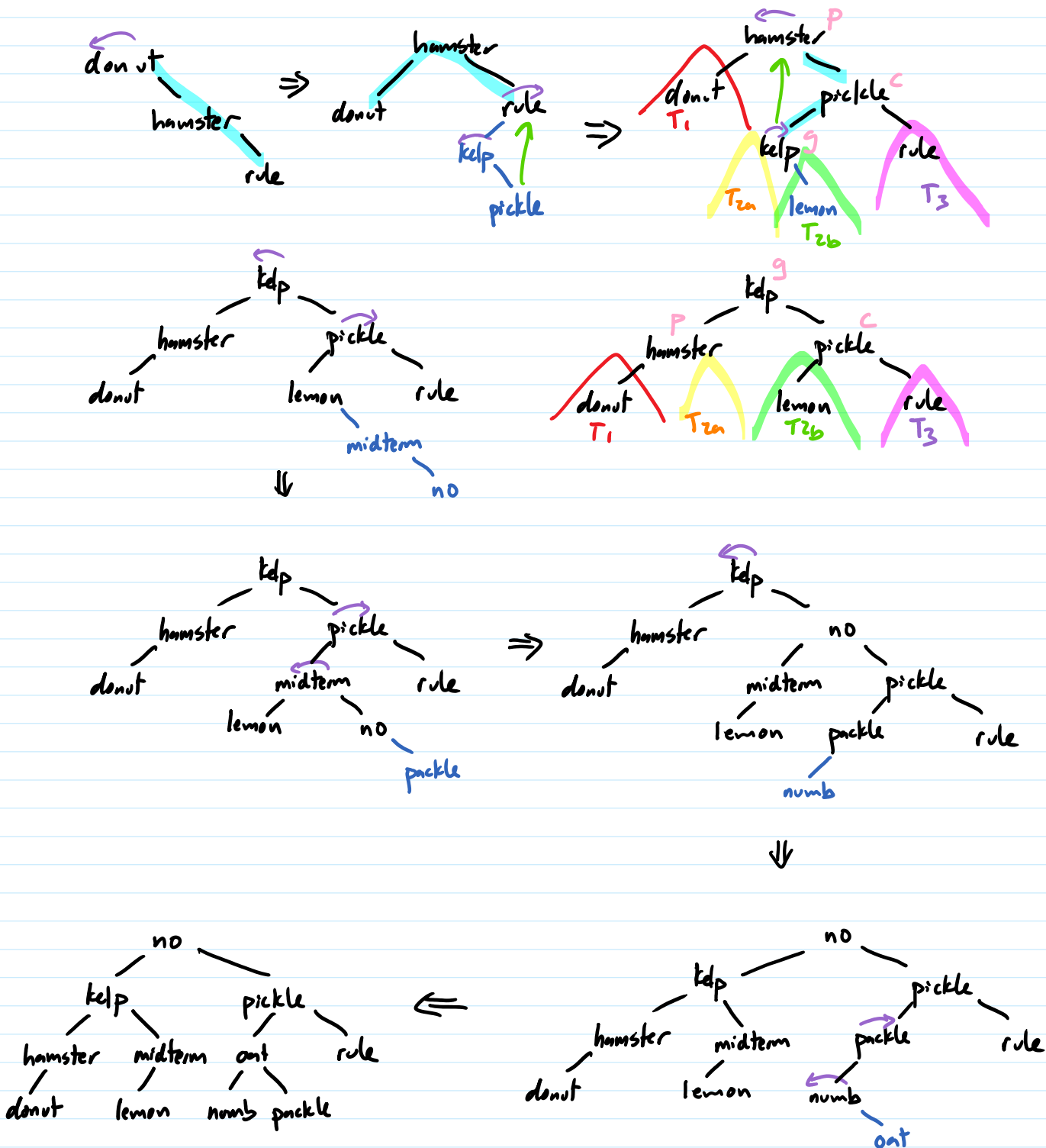
Double Rotation Example



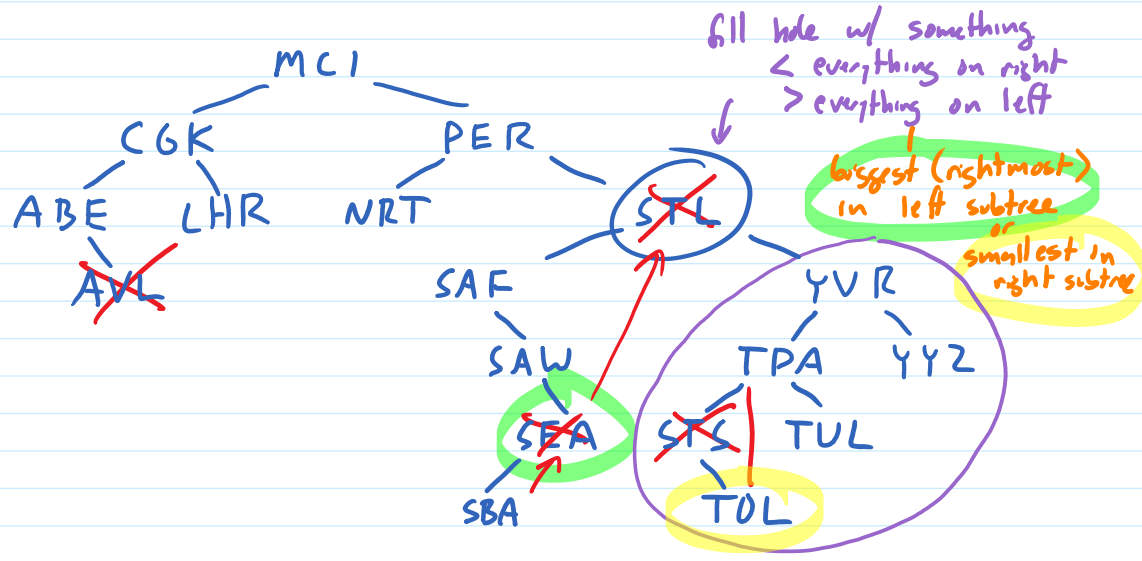
ACC CAI DXB EDI FAI
 FCD ICN JFK PVD

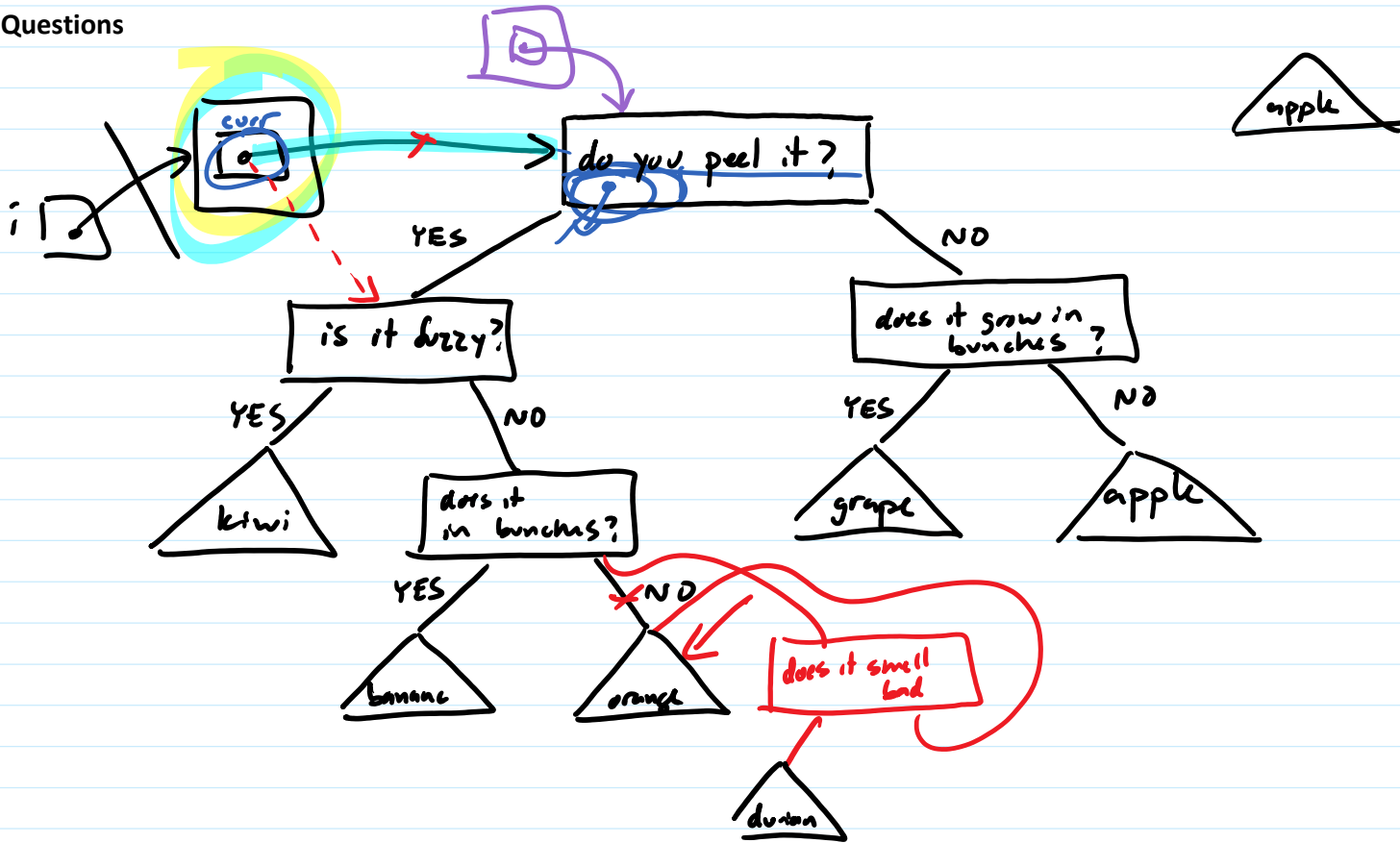
AVL Example

donut hamster rule kelp pickle lemon midterm no packle numb oat

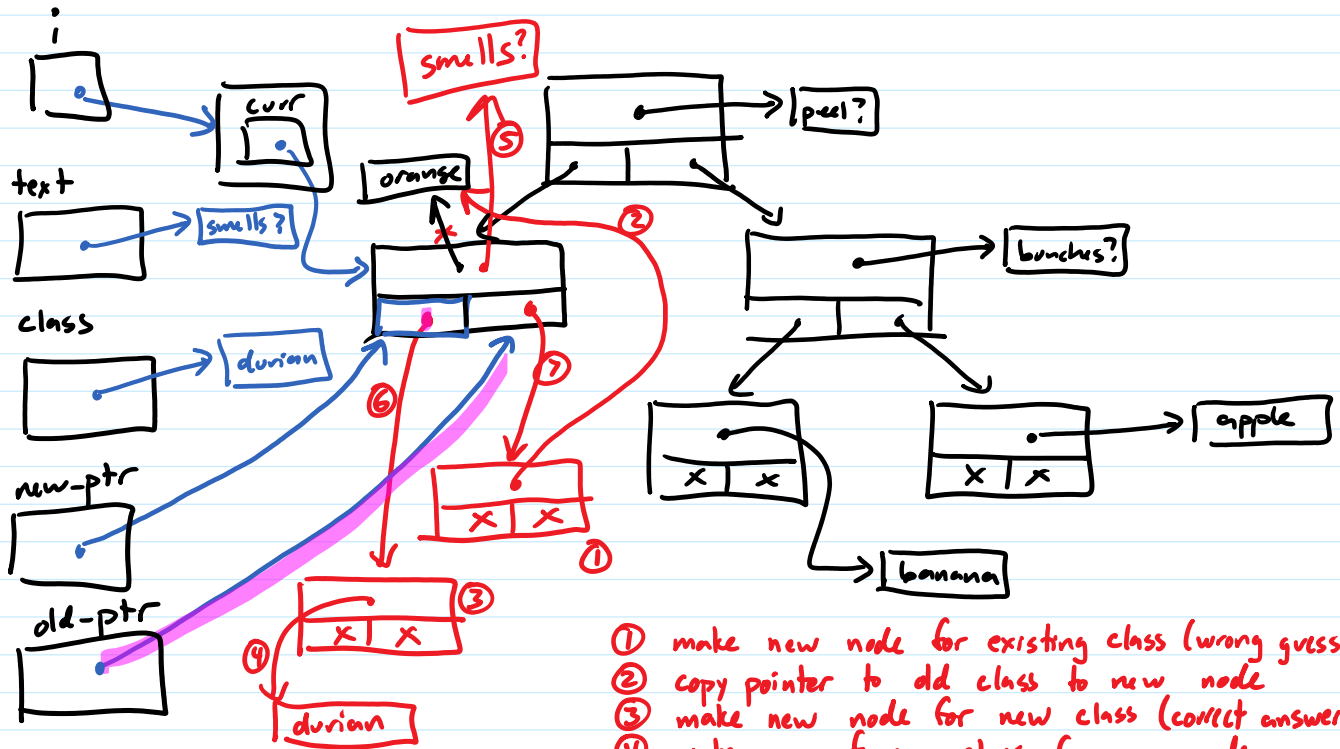


Removing from a BST





Add branch



- ① make new node for existing class (wrong guess)
- ② copy pointer to old class to new node
- ③ make new node for new class (correct answer)
- ④ make copy of new class for new node
- ⑤ replace text in existing node with copy of new test (new?)
- ⑥ point pointer new_ptr points to to node for new class
- ⑦ point pointer old_ptr points to to new node for old class