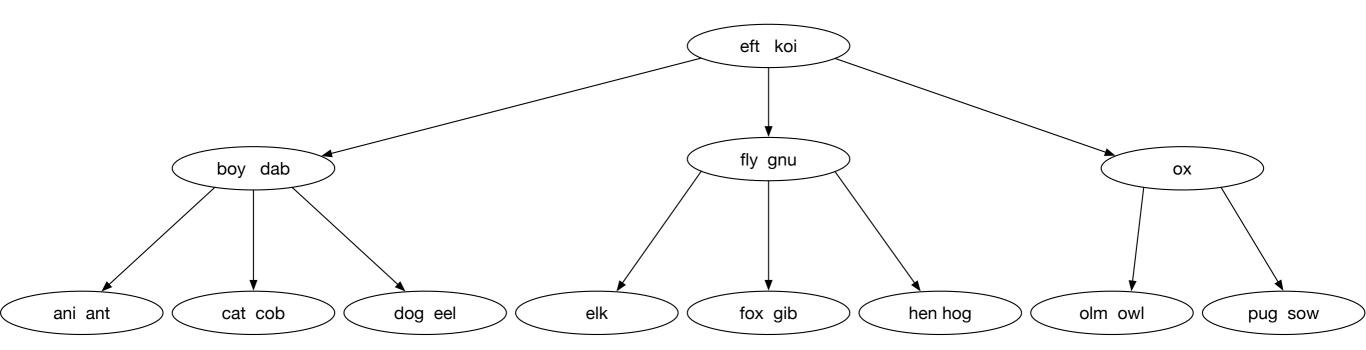
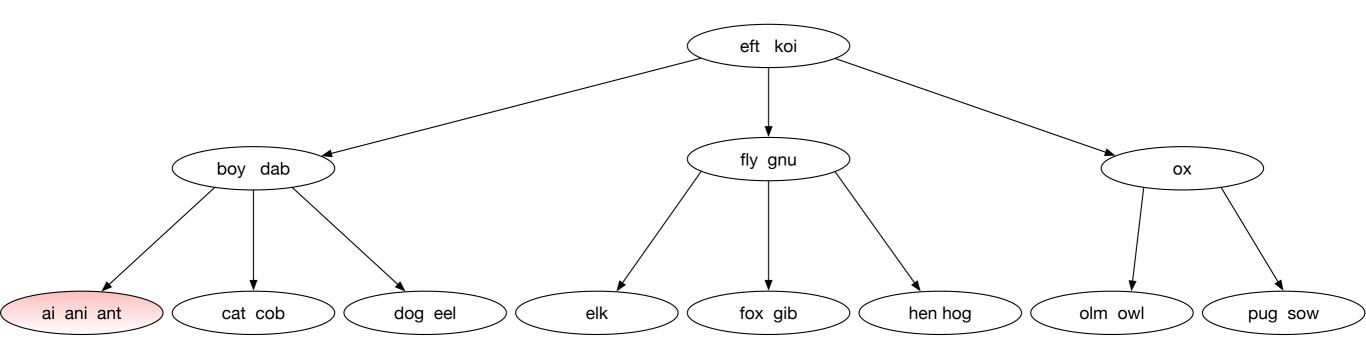
Activity: B-tree Operations

How do we insert 'ai'?

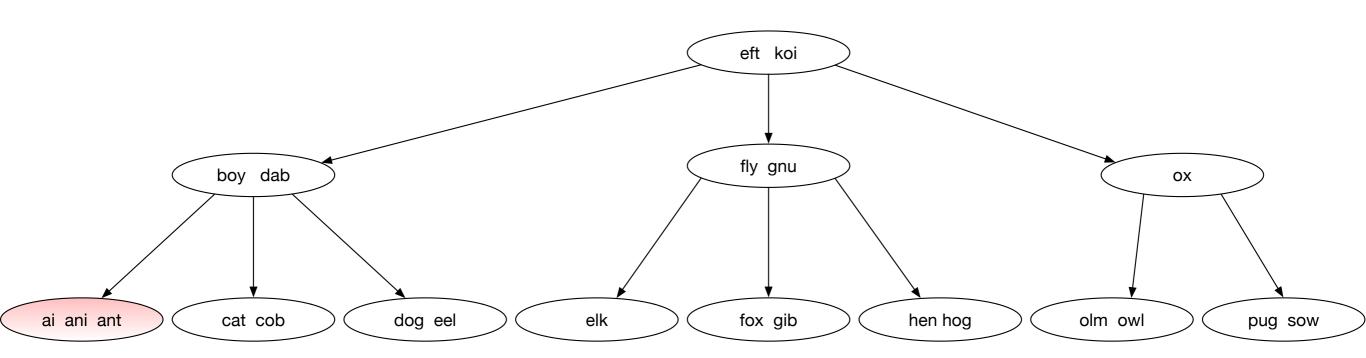
- Rules:
 - Prefer left rotate over right rotate over split / merge



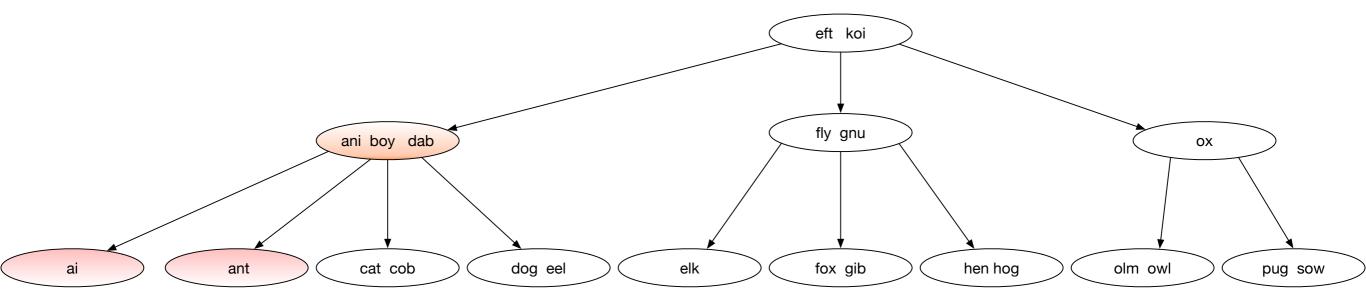
Insert into leaf and diagnose an overflow



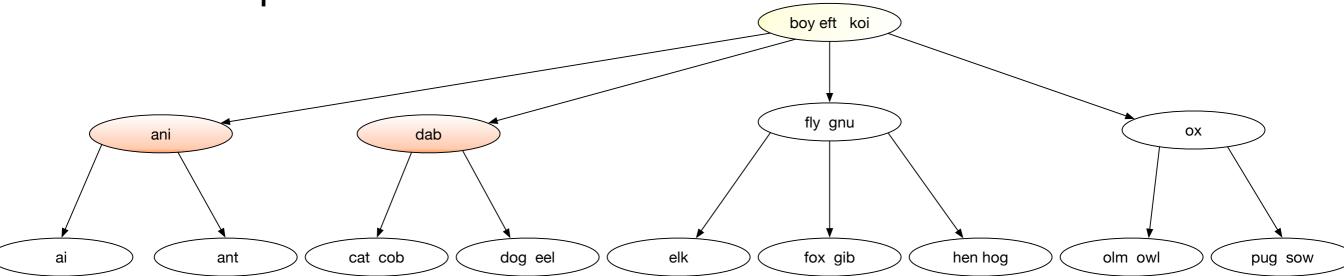
• Only neighbor is at capacity, need to split



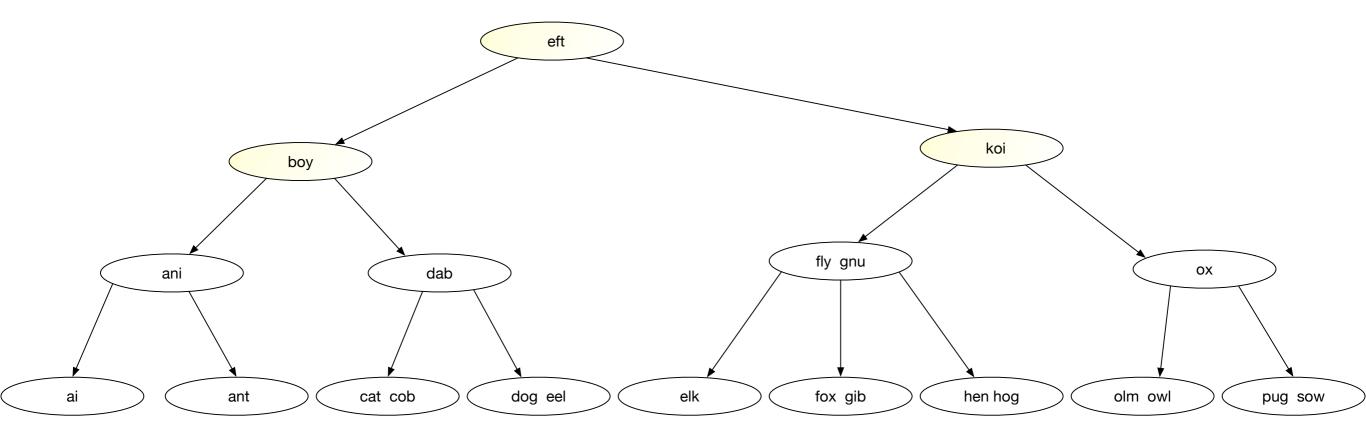
The split results in another overflow



- The only neighbor is at capacity, therefore split
 - The parent is at capacity, has no neighbor, so we need to split.

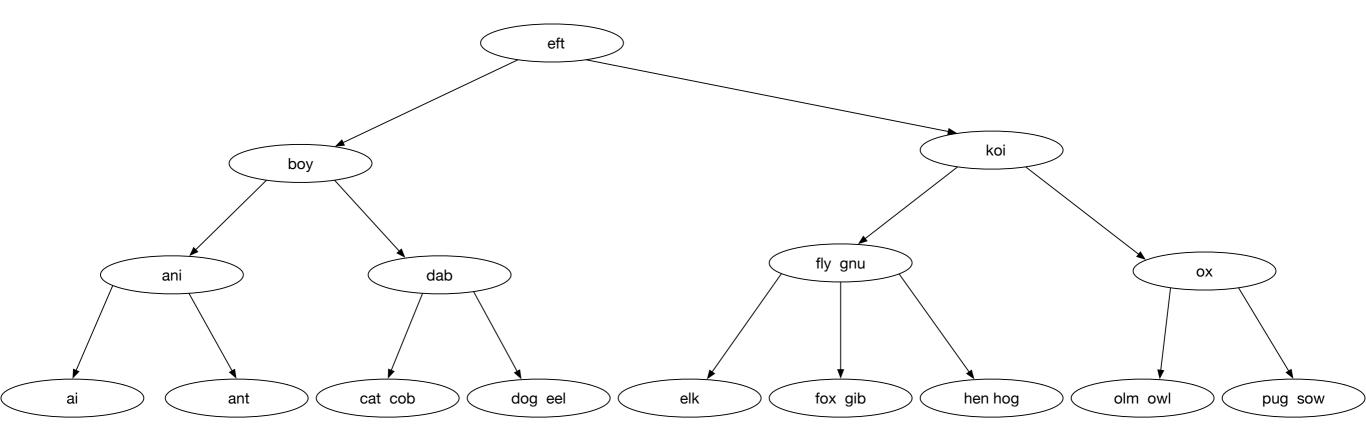


Final result



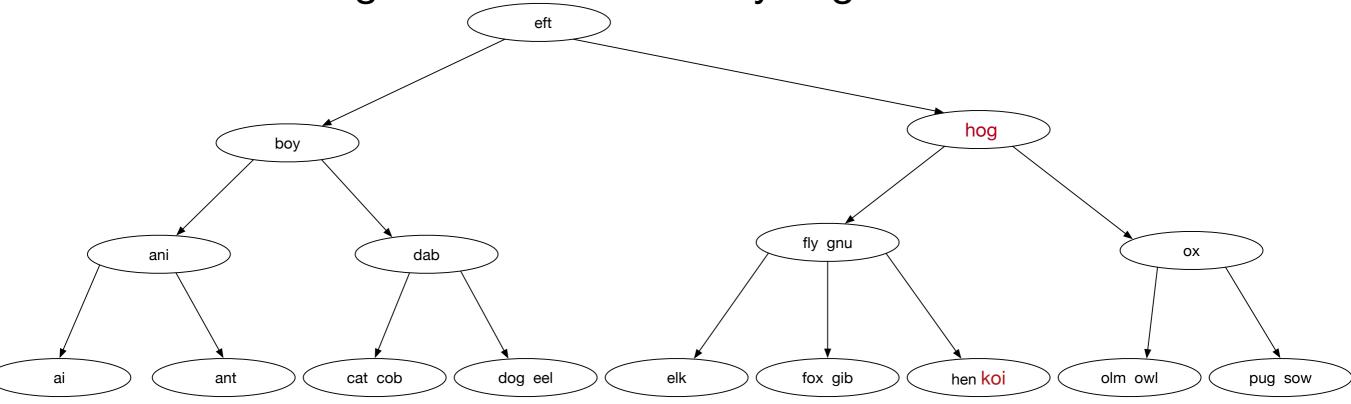
Problem

- Delete 'koi' from this tree
 - Rules: Use always the predecessor
 - Prefer left rotate over right rotate over merge

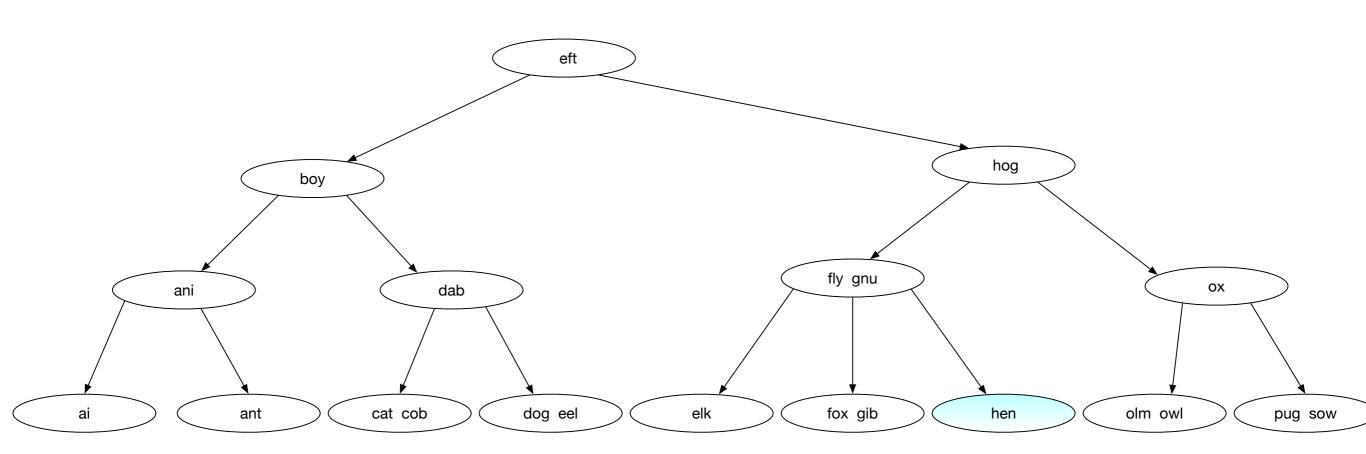


Locate "koi", then the predecessor of "koi" and switch

From "koi" go left and then always right

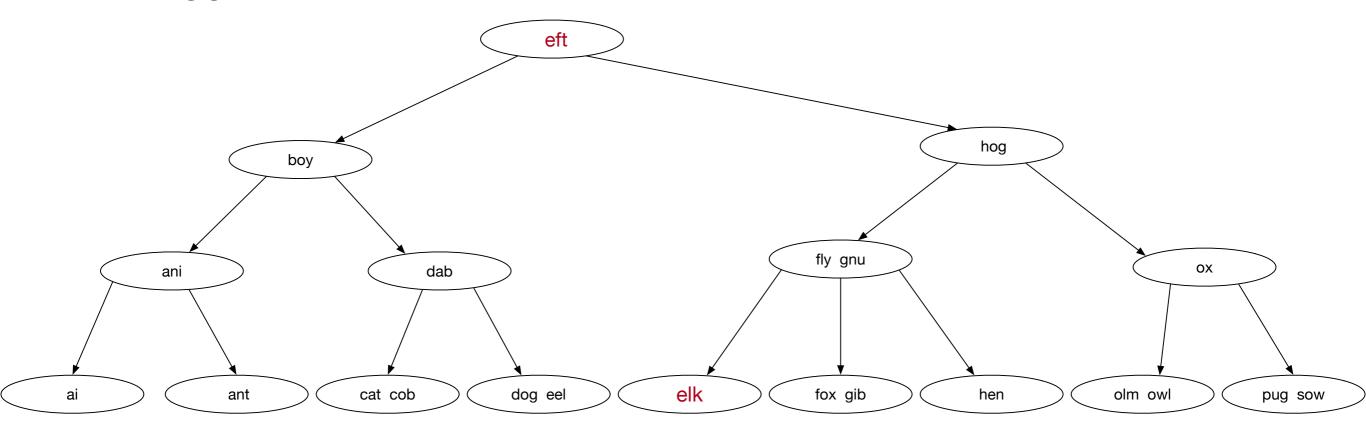


 The deletion does not lead to an underflow, so we have the final state of the 2-3 tree

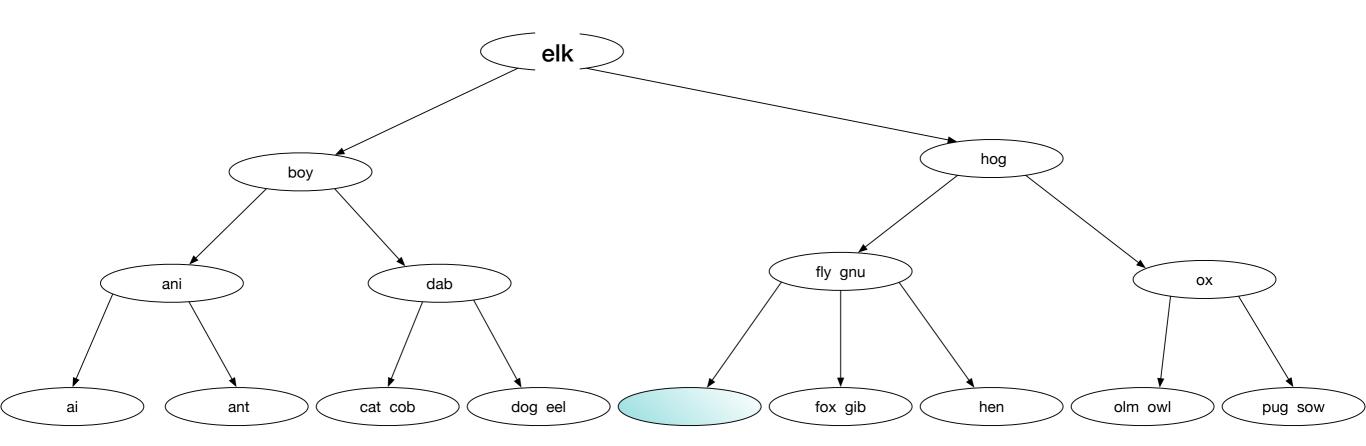


Problem

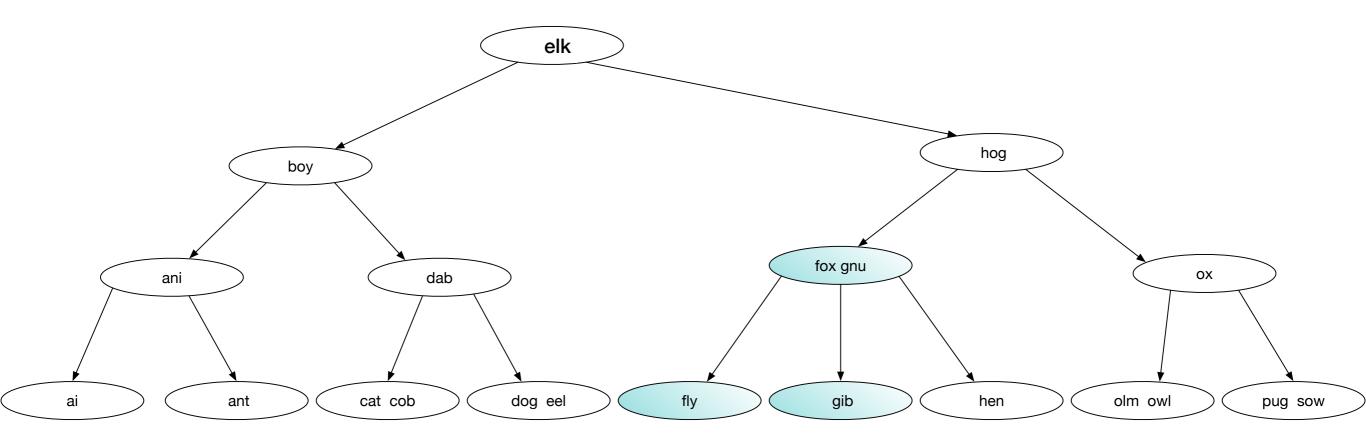
 Now use the successor to delete "eft" from the previous tree



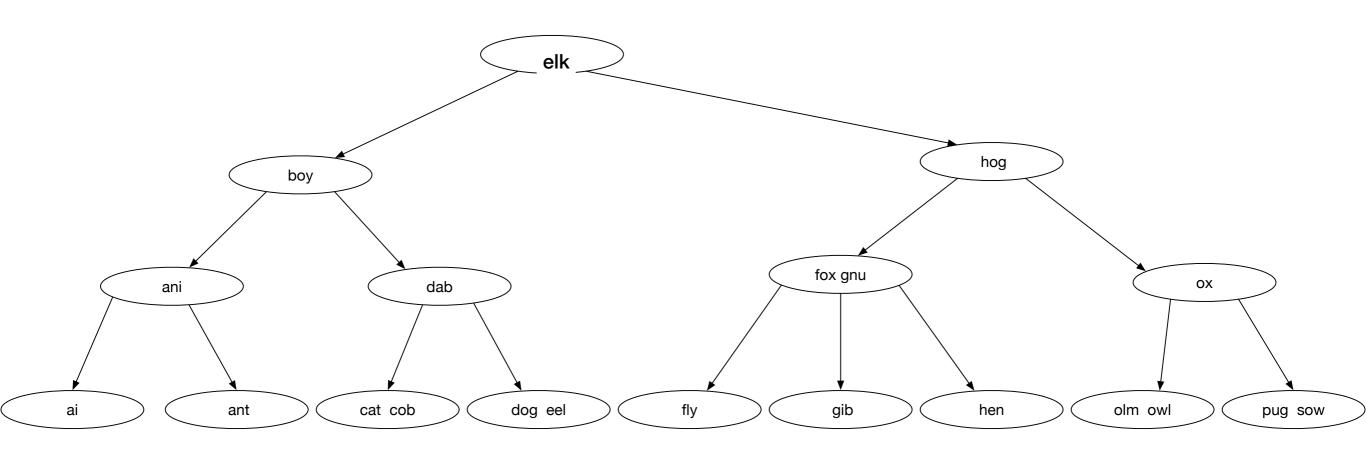
• Now the delete gives an empty (underflowing) node



- There is only one neighbor, which is over minimum capacity, so we rotate
- "fox" goes up, "fly" goes down

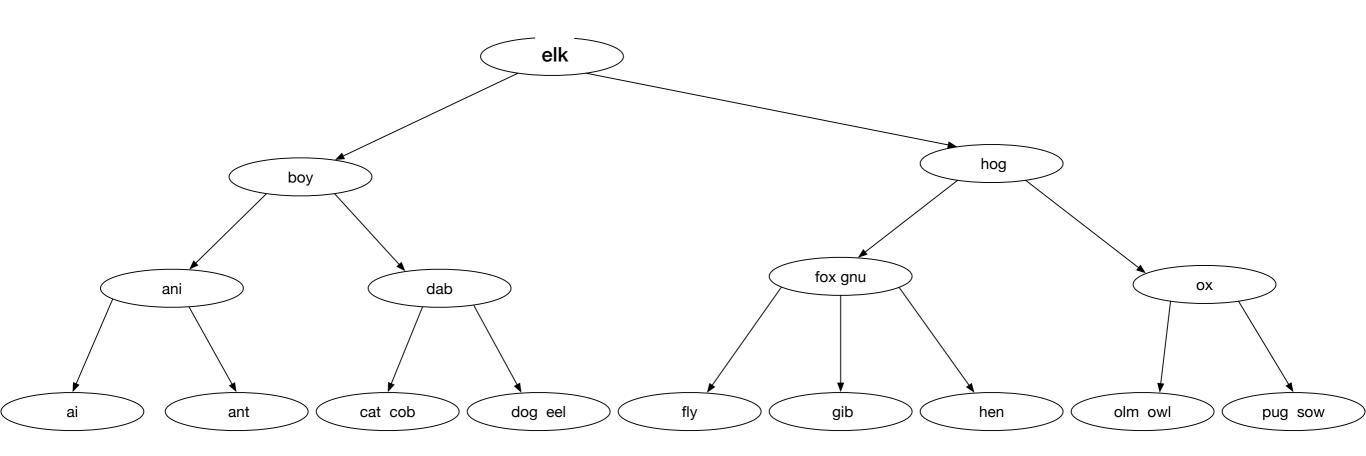


• The result is a valid 2-3 tree

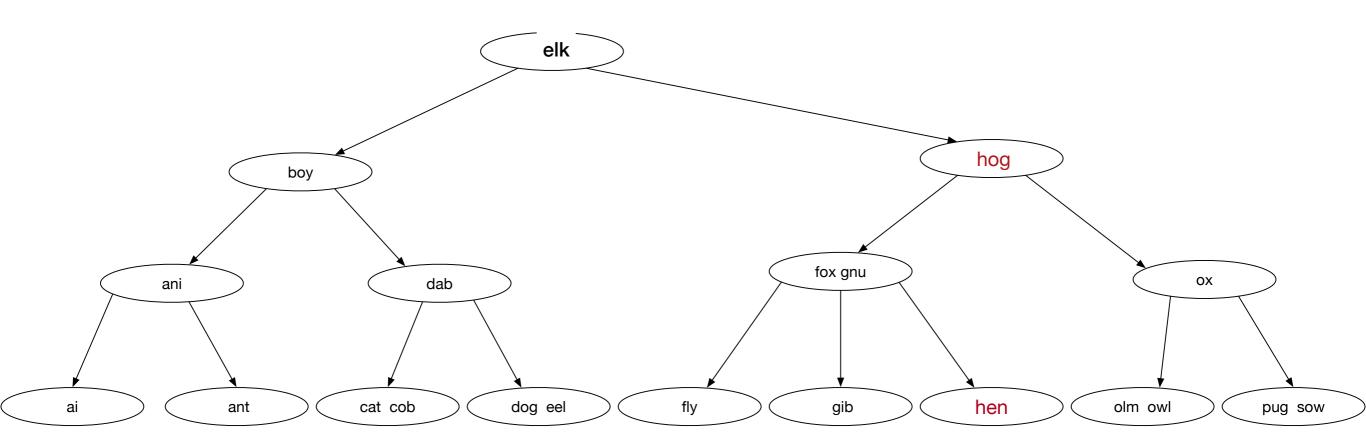


Problem

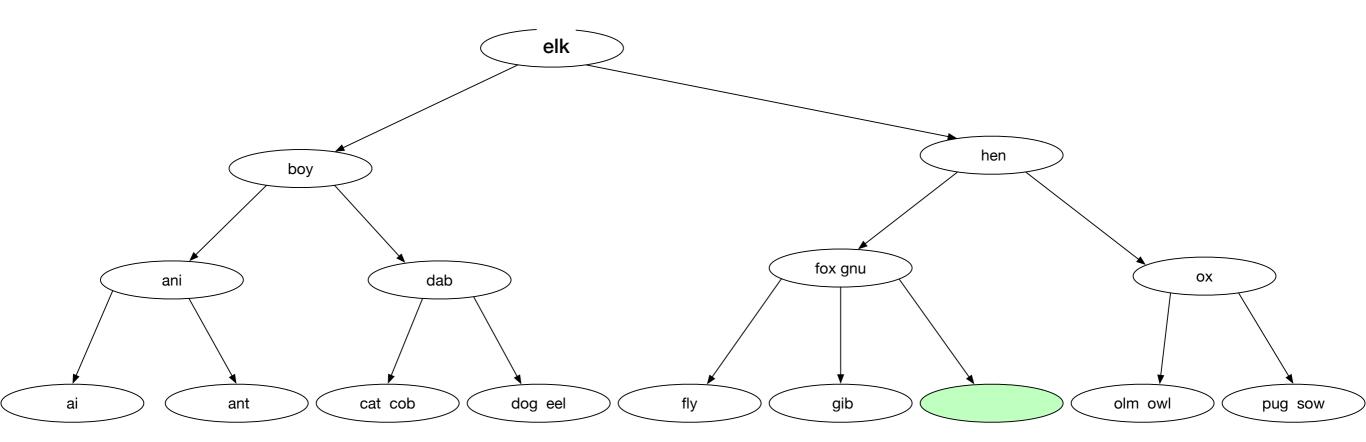
Delete hog using the predecessor



We find 'hog', then we locate the predecessor

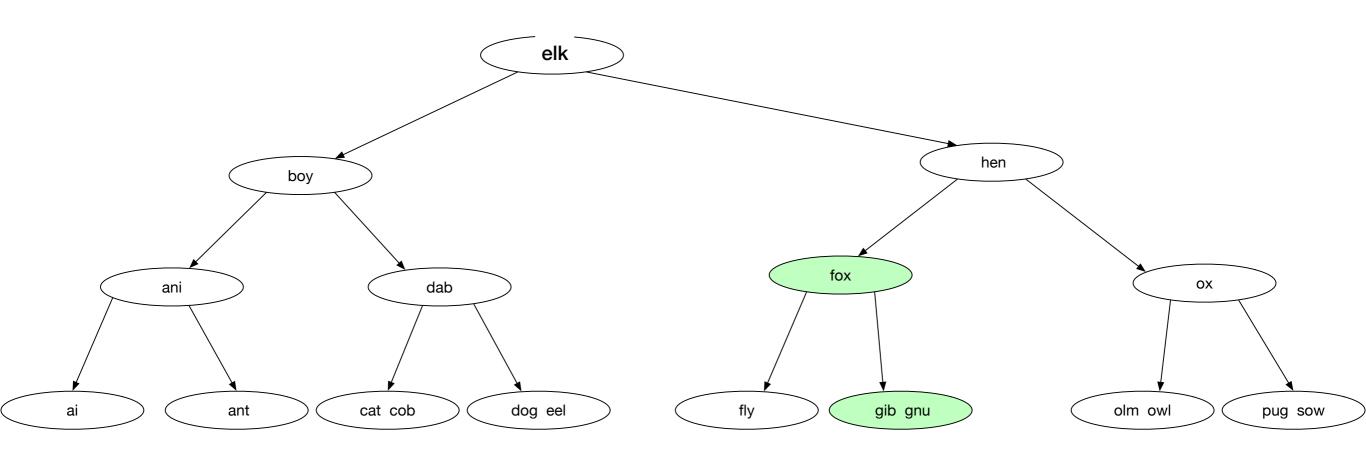


- We switch and delete from the leaf
- The result is an empty node



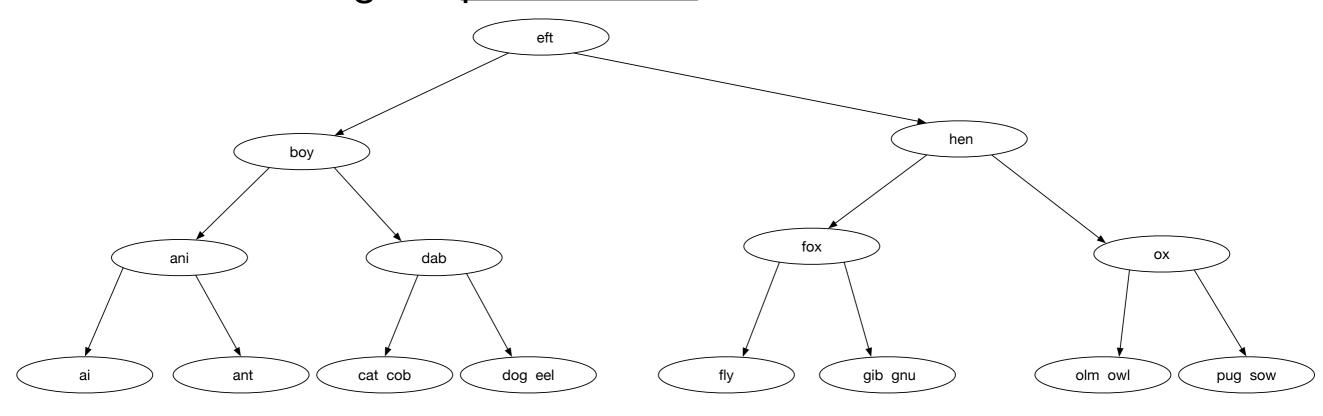
- Because we cannot do a rotate, we need to do a merge
- A merge is easiest remembered as the inverse operation of the split
 - "gnu" goes down into the united node

 The resulting tree complies with all requirements and we are done



Homework Problem

- Submit by Wednesday:
 - Show the steps and the result of the deletion of "eft" using the <u>successor!</u>
 - Then show the steps and the result of the deletion of "fox" using the <u>predecessor!</u>



Rules

- Electronic submission via D2L ONLY
- Formatted
- Scanned images not accepted