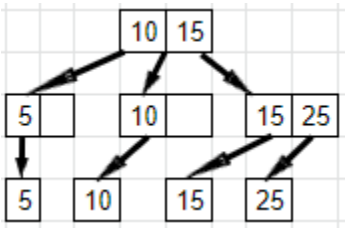
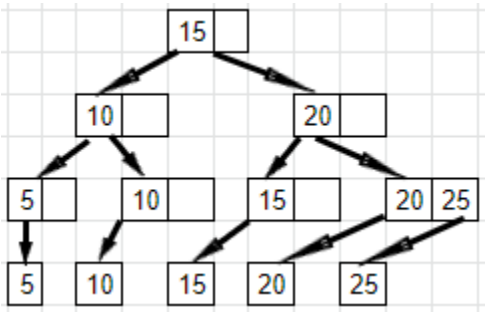
Adrien Protzel | protzela | protzela@oregonstate.edu
Curtiss Haldorson | haldorsc | haldorsc@oregonstate.edu

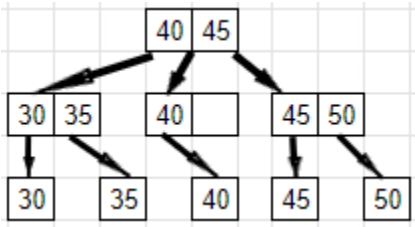
1.

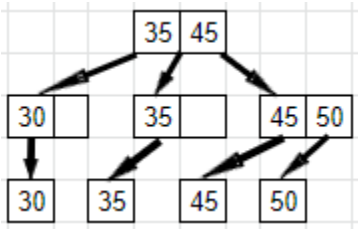
a. 2d*8+(2d+1)*4<=30 d<=13/12
With inserting a 20, the 15|25 block needs to be split. Once split, 10|15 can only point to 3 blocks and so a new block is made to point to 20|25





b. 2d*4+(2d+1)*4<=30 d<=13/8
With deleting 40, the middle block is empty and needs to be filled. The 30|35 block on the right is split to fill it. The 40|45 is updated to point to the 30| and 35|.





2.

a. The ideal tree to minimize the time spent on this specific query would be:
composite(because each index has multiple values),
clustered(aka sorted, because this way the algorithm can ignore large sections of the tree that are irrelevant to the search instead of parsing through the whole thing index by index),
sorted first by age(because this specific query is looking for only a single age, which allows the relevant part of the tree to be identified),
and then by salary(for which it would be easy to simply grab the pre-sorted range)
Aka: the key should be <age, salary>

3.

a. Separate submission.