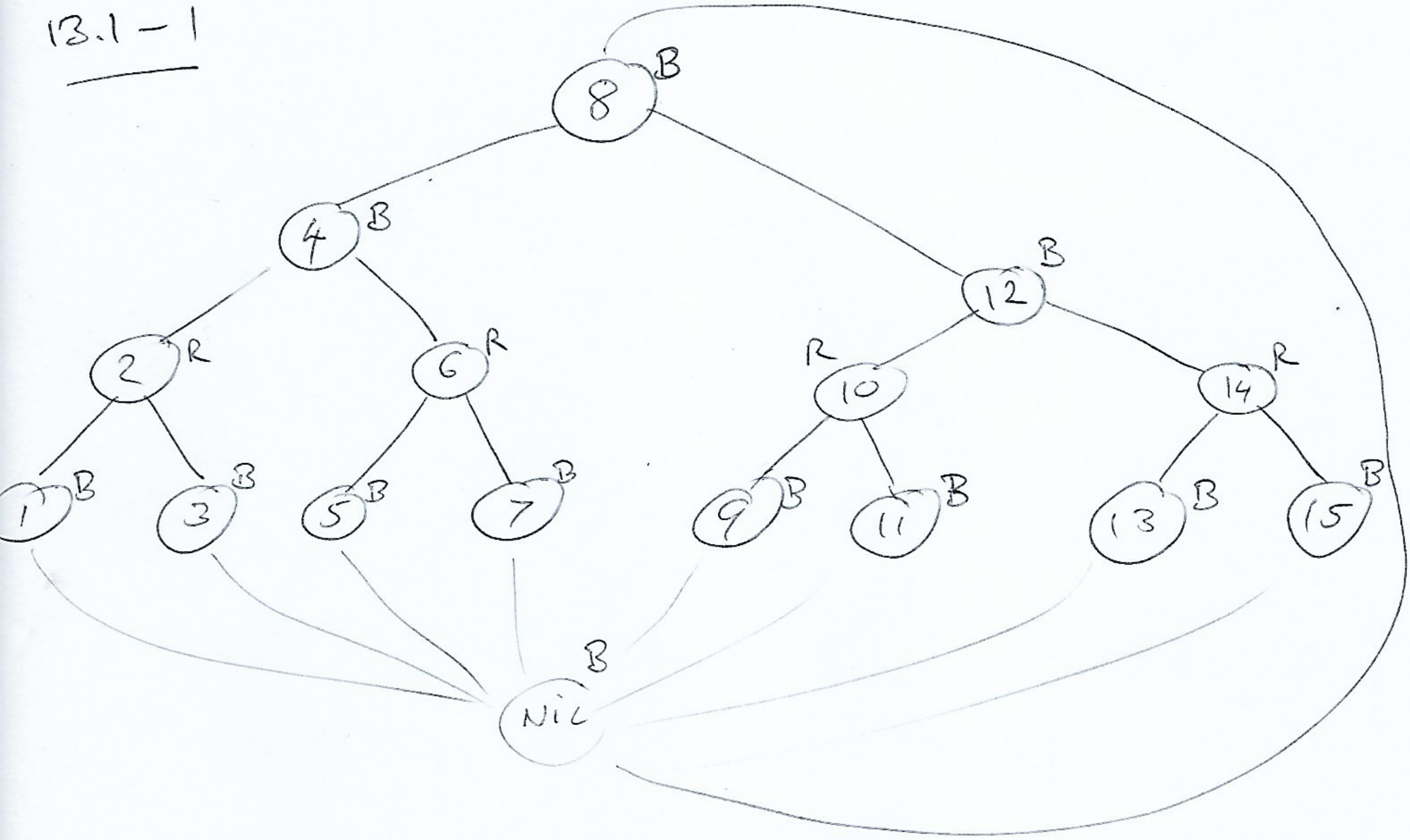
13.1 - 1

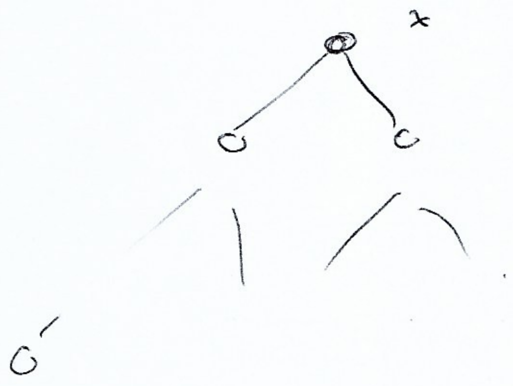


The above tree has black-height (at root) equal to 3.

For black-height 4, all nodes must be B.

For black-height 2, alternate B - R - B - R - B on the different levels.

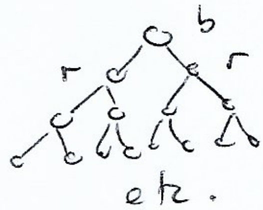
13.1 - 5



The black-height of x is the number of black nodes in any simple path from x to a leaf. The shortest possible path has length $bh(x)$, if it has no red nodes. The longest possible path alternates between black and red nodes by rule 4. of red-black trees, so its length is $2\,bh(x)$.

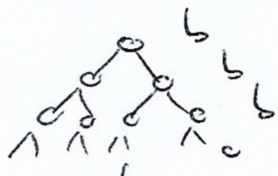
Thus, the longest path is less or equal to twice the shortest path.

13.1 - 6 The largest number of internal nodes is $2^{2k}-1$:



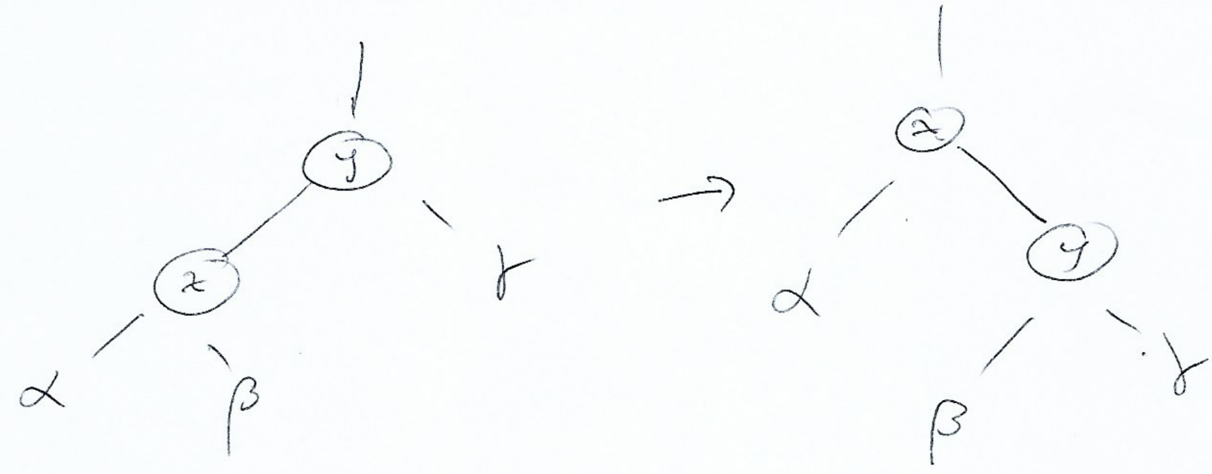
b - all black
r - all red

The smallest number of internal nodes is $2^{k}-1$:



all black

13.2 - 3



If a is a node in α, then its depth decreases by 1.

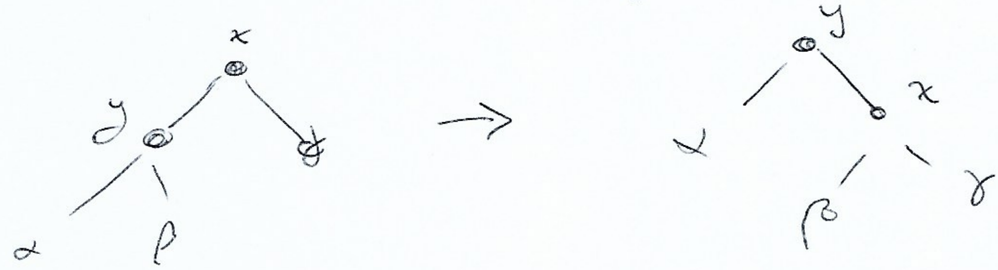
If b is a node in β, then its depth is unchanged.

If c is a node in γ then its depth increases by 1.

13.2 - 4

We first prove the Hint: At most $n-1$ right rotations are needed to transform a tree into a right-going chain, for an n-node tree.

The method is to right-rotate at the root while the left-subtree is not empty.
Note that each such right-rotation places a new node on the right-subchain:



Both x & y on right subchain.

When the left-subtree of the root is empty, repeat the process at its right child.
Again, each right rotation places a new node on the right subchain.
Repeat until tree is a right-going chain.
At the start there was at least one node on the right-subchain - the root node.
Thus, we need at most $n-1$ right-rotations

For the main question, suppose we want to transform tree T into tree T'.

First, transform T into a right-going chain using at most n-1 right-rotations.

Next, we can separately transform T' into a right-going chain.

Then, to the right-going chain obtained from T we apply left-rotations in the reverse order of the right-rotations we did to transform T' into a right-going chain.