Height of a Red-Black Tree is $O(\log n)$

We showed that the height, h, of a *Red-Black Tree* is $O(\log n)$.

Recall that we used $\mathcal{B}$ to denote the number of *Black* nodes on a path in the RB-Tree and that all possible paths starting at the root must have $\mathcal{B}$ Black nodes. (Keep in mind that $\mathcal{B}$ *is not* the total number of Black nodes in the tree, just on any path.)

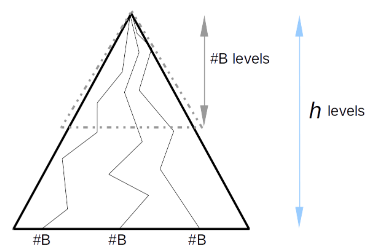
The proof was based on the following claims:

- Claim 1: A *perfect binary tree* of X levels has $2^X - 1$ nodes. (This was shown earlier in the course; see the relevant notes.)
- Claim 2: $\mathcal{B} \leq h \leq 2\mathcal{B}$

  The first part, $\mathcal{B} \leq h$, is due to the fact that the RB-Tree needs to have at least $\mathcal{B}$ levels if every possible path has to have at least $\mathcal{B}$ black nodes. But the total height, h, can be longer, since some paths might have Red nodes.

  The second part, $h \leq 2\mathcal{B}$, is due to the fact that the longest possible path in RB-Tree has alternating Black and Red nodes starting with Black (the root), i.e. $\mathcal{B} = \mathcal{R}$ on longest possible path, so the longest possible path has $\mathcal{B} + \mathcal{R} = 2\mathcal{B}$ nodes/levels. Since the tree cannot be taller than longest possible path, we have $h \leq 2\mathcal{B}$.

The key point in the proof was to find a *perfect binary tree* inside a given RB-Tree.



Given RB-Tree (the solid triangle in the figure) of n nodes, h levels, and $\mathcal{B}$ Black nodes on each possible path, we observed that (i) the dashed triangle in the figure must have $\mathcal{B}$ levels and (ii) those levels must be fully packed. There should be no gaps in the dashed triangle, since otherwise there will be a path that is shorter than the $\mathcal{B}$ levels, and therefore, it will not be possible to have the required $\mathcal{B}$ Black nodes on that path.

Note that we do not claim that the dashed triangle has only Black nodes. We just claim that the dashed triangle is completely packed with nodes, thus it is a *perfect binary tree*.

Since the dashed triangle of $\mathcal{B}$ levels is a *perfect binary tree*, we can use Claim 1 to establish that it has $1 + 2^1 + 2^2 + 2^3 + ... + 2^{\mathcal{B}-1} = 2^{\mathcal{B}} - 1$ nodes. Again, this was proven earlier in the course; see the relevant notes.

Since the *perfect subtree* (the dashed triangle) is part of the given RB-Tree (the solid triangle), the number of nodes, n, in the given RB-Tree may be more than the number of nodes in the *perfect subtree*, so we have:

$$2^{\mathcal{B}} - 1 \leq n$$

$$2^{\mathcal{B}} \leq n + 1$$

$$log_2 2^{\mathcal{B}} \leq log_2(n+1)$$

$$\mathcal{B}\ log_2 2 \leq log_2(n+1)$$

$$\mathcal{B} \leq log_2(n+1)$$

Now we use the second part of Claim 2, i.e. $h \leq 2\mathcal{B}$, where h is the height of the given RB-Tree that has $\mathcal{B}$ nodes on each path. We just showed that:

$$\mathcal{B} \leq log_2(n+1)$$

so multiplying by 2 we get:

$$2\mathcal{B} \leq 2log_2(n+1)$$

and since $h \leq 2\mathcal{B}$ we have:

$$h \leq 2\log_2(n+1)$$

Therefore, h is $O(\log n)$.