In class today I gave an inductive proof that always merging smaller trees into larger in the union-find data structure keeps the trees balanced. As pointed out to me by Kun He, the proof was not correct, because I left out a case. Here is a correct proof.

**Lemma 1.** In the union-find data structure, if we always merge smaller trees into larger (that is, make the root of the smaller tree point to the root of the larger), then for any tree $t$ at any point in time,

$$\text{size}(t) \geq 2^{\text{height}(t)}.$$  

**Proof.** By induction on the number of union operations. For the basis, initially all trees are single nodes, and $\text{size}(t) = 1 = 2^0 = 2^{\text{height}(t)}$.

For the induction step, suppose at some point we merge $t_2$ into $t_1$ to get $t_3$. There are two cases:

**Case 1.** The height of $t_1$ does not grow by merging $t_2$ into it; that is, $\text{height}(t_3) = \text{height}(t_1)$. Then

$$\begin{align*}
\text{size}(t_3) &= \text{size}(t_1) + \text{size}(t_2) \\
&\geq \text{size}(t_1) \\
&\geq 2^{\text{height}(t_1)} \quad \text{induction hypothesis} \\
&= 2^{\text{height}(t_3)} \quad \text{since } \text{height}(t_3) = \text{height}(t_1).
\end{align*}$$  

**Case 2.** The height of $t_1$ grows by merging $t_2$ into it; that is, $\text{height}(t_3) = \text{height}(t_2) + 1$. Since we always merge smaller trees into larger, $\text{size}(t_2) \leq \text{size}(t_1)$. Then

$$\begin{align*}
\text{size}(t_3) &= \text{size}(t_1) + \text{size}(t_2) \\
&\geq 2 \cdot \text{size}(t_2) \quad \text{since } \text{size}(t_2) \leq \text{size}(t_1) \\
&\geq 2 \cdot 2^{\text{height}(t_2)} \quad \text{induction hypothesis} \\
&= 2^{\text{height}(t_2)+1} \\
&= 2^{\text{height}(t_3)} \quad \text{since } \text{height}(t_3) = \text{height}(t_2) + 1.
\end{align*}$$