Thomas Haddy
Section 9 @ 11:00 Th
TA: Trent Muhr
1/25/18

COM S 311 HW 1

1. Let the Fibonacci numbers be defined recursively as follows:

$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-1} + F_{n-2}$

Prove $P(n)$, the following property of Fibonacci numbers:

$F_1^2 + F_2^2 + F_3^2 + \ldots + F_n^2 = F_n \cdot F_{n+1}, \forall n \geq 1$

Proof:

Base: Let $n=1$, $F_1 = 1 \rightarrow F_1^2 = 1$. $F_{(1)} \cdot F_{(1)+1} = 1 \cdot (1+0) = 1$

Since $F_1^2 = 1$ and $F_n \cdot F_{n+1}$ are equal, $P(1)$ holds true.

Inductive Hypothesis: Assume $P(k)$: $F_1^2 + F_2^2 + \ldots + F_k^2 = F_k \cdot F_{k+1}$, $\forall k \geq 1$. Prove $P(k+1)$: $F_1^2 + F_2^2 + \ldots + F_k^2 + F_{k+1}^2 = F_{k+1} \cdot F_{k+2}$, $\forall k \geq 1$

Inductive step: $[F_1^2 + F_2^2 + \ldots + F_k^2] + F_{k+1}^2 = F_{k+1} \cdot F_{k+2}$

$F_k \cdot F_{k+1} + F_{k+1}^2 = F_{k+1} \cdot F_{k+2}$ by I.H.

$F_{k+1}[(F_k + F_{k+1})] = F_{k+1} \cdot F_{k+2}$

$F_{k+1} \cdot F_{k+2} = F_{k+1} \cdot F_{k+2}$ by recursive def. of Fibonacci numbers

Therefore by mathematical induction, since $P(1)$, $P(k)$, and $P(k+1)$ are true, $P(n)$ is true $\forall n \geq 1$.

2\. a) Defining $n(T)$: The vertices of T are the root of T and the vertices of the two subtrees T_1 and T_2. The number of the nodes in a tree T is $n(T)$, where

Base: $n(T) = 1$ if T consists of only a root r

Recursive step: $n(T) = 1 + n(T_1) + n(T_2)$ if $T = T_1 \cdot T_2$

Defining $h(T)$: The height of T is $h(T)$.

Base: if T is a single node, then $h(T) = 1$

Recursive step: if T consists of root r and subtrees T_1 and T_2, then

$$h(T) = 1 + \max(h(T_1), h(T_2))$$

Proof: For every FBT T, prove $n(T) \geq 2h(T) - 1$

Base: Let T consist of 1 node that is the root.

So $n(T) = 1$ and $h(T) = 1 \rightarrow 2(1) - 1 = 1$. So $1 \geq 1$.

Therefore the base case holds true.

Inductive Hypothesis: Let T_1 and T_2 be two FBT's with heights h_1 and h_2. Assume that $n(T_1) \geq 2h(T_1) - 1$ and $n(T_2) \geq 2h(T_2) - 1$. Also, assume $h_1 \geq h_2$.

Inductive Step: Let T be a new tree with r as a root and T_1 and T_2 being left and right subtrees. Let h be its height. Prove that $n(T) \geq 2h(T) - 1$.

So, $n(T) = n(T_1) + n(T_2) + 1$ by rec. step of $n(T)$

$n(T) \geq (2h(T_1) - 1) + (2h(T_2) - 1) + 1$ by I.H.

$n(T) \geq 2h(T_1) + 2h(T_2) - 1$

$n(T) \geq 2(h(T_1) + h(T_2) + 1) - 1$

$n(T) \geq 2h(T) - 1$ by rec. def. of $h(T)$

Therefore, every FBT T, $n(T) \geq 2h(T) - 1$ is true.

2. b) Defining $n(T)$: same as part a)

Defining $i(T)$: The number of internal nodes in T.

Base: if T is a single root r, then $i(T)=0$

Recursive step: $i(T) = 1 + i(T_1) + i(T_2)$ if T consists of root r and subtrees T_1 and T_2.

Proof: For every FBT T, prove $i(T) = (n(T)-1)/2$

Base: Consider a FBT having a single node. So $i(T)=0$.

$n(T) = 1 \rightarrow ((1)-1)/2 = 0$. $0 = (1-1)/2 = 0$ ✓

Therefore the base case holds true.

Inductive Hypothesis: Let T_1 and T_2 be 2 FBT's.

Assume that $i(T_1) = (n(T_1)-1)/2$ and $i(T_2) = (n(T_2)-1)/2$.

Inductive Step: Consider the FBT T with a new root, whose children are the roots of T_1 and T_2. Prove the following:

$i(T) = i(T_1) + i(T_2) + 1$ by rec. def. of $i(T)$

$i(T) = \dfrac{n(T_1)-1}{2} + \dfrac{n(T_2)-1}{2} + 1$ by I.H.

$2i(T) = n(T_1) - 1 + n(T_2) - 1 + 1$

$2i(T) = [n(T_1) + n(T_2) + 1] - 2$

$2i(T) = n(T) - 2$ by rec. def. of $n(T)$

$i(T) = \dfrac{n(T)-1}{2}$

Therefore, every FBT T, $i(T) = \dfrac{n(T)-1}{2}$ is true

3. Proof:

Base: Let the i^{th} iteration be 0. $a^n = x_0^{m_0} \cdot y_0$ where $x_0 = a, m_0 = n,$ and $y_0 = 1$. So $a^n = a^n \cdot 1 \to a^n = a^n$ ✓

Therefore, the base case holds true.

Inductive Hypothesis: Assume $\forall k$, $a^n = x_k^{m_k} \cdot y_k$.

Prove $a^n = x_{k+1}^{m_{k+1}} \cdot y_{k+1}$ where $k+1$ is the next iteration.

Inductive Step: There will be 2 cases to prove $P(k+1)$ holds and that is if m is odd or even.

Case 1: (m is odd)

$x_{k+1} = x^2$, $m_{k+1} = (m-1)/2$, $y_{k+1} = x$ by looking at program

$$a^n = x_{k+1}^{m_{k+1}} \cdot y_{k+1}$$

$$a^n = x^2 \cdot x_k^{(m-1)/2 \cdot m_k} \cdot x \cdot y_k$$

$a^n = a^n \cdot x^{2[(m-1)/2]} \cdot x$ by I.H.

So the next iteration is what we expect by sub.

$$a^{n+1} = a^n \cdot x_{k+1}^{m_{k+1}} \cdot y_{k+1} \to a^{n+1} = a^{n+1}$$

Case 2: (m is even)

$x_{k+1} = x^2$, $m_{k+1} = m/2$, $y_{k+1} = 1$ by looking at program

$$a^n = x_{k+1}^{m_{k+1}} \cdot y_{k+1}$$

$a^n = x^2 \cdot x_k^{(m/2)m_k} \cdot 1 \cdot y_k \to a^n = a^n \cdot x^{2(m/2)} \cdot 1$ by I.H.

$$a^{n+1} = a^n \cdot x^m \to a^{n+1} = a^{n+1}$$

Therefore, $\forall i$ $a^n = x_i^{m_i} \cdot y_i$ holds true.

4. Base: Let $i=0$ for the 0^{th} iteration.

left $=0$, right $= n-1$ where n is the length of array and $n \geq 1$

So $0 \leq 1$ ✓ Therefore the base case holds true.

Inductive Hypothesis: Assume $left_k \leq right_k$ where k is the k^{th} iteration. Also assume there is a T in the array such that two ints in the array equal T. Prove $left_{k+1} \leq right_{k+1}$.

Inductive Step: This will need to be split into 3 cases where $k+1$ yields true, left++, or right--.

Case 1: $(x=T)$ Assume $left_k \leq right_k$. So on the next iteration the left and right variables will be unchanged and therefore equal to its previous iteration.
So $left_k = left_{k+1}$ and $right_k = right_{k+1}$. This means that $left_{k+1} \leq right_{k+1}$.

Case 2: $(x < T)$ Assume $left_k \leq right_k$. If left = right from the k^{th} iteration, then on the $k+1^{th}$ iteration where $x < T$, left becomes left > right but this case never happens due to the I.H. which states we assume T exists. So left is never > right. If left < right from the k^{th} iteration, then on the $k+1^{th}$ iteration where $x < T$, left++ so left is either left $\leq$ right.

Case 3: $(x > T)$ Assume $left_k \leq right_k$. If left = right from the k^{th} iteration, then at $k+1^{th}$ iteration where $x > T$, right would become < left but this never happens because the I.H. states there exists a T. So right is never < left. If right > left from the k^{th} iteration, then on the $k+1^{th}$ iteration where $x > T$, right -- so right is either right $\geq$ left.

4. (Cont.) Therefore since all 3 cases are true along with the base case, we can say that $left_{k+1} \leq right_{k+1}$ for $\forall k \geq 0$ assuming there exists a T such that $x = T$.

Proof p2: Prove $\exists$ indices l and j s.t. $a[l] + a[j] = T$, then $left \leq l \leq j \leq right$. Assume $a[l]$ and $a[j] > 0$

Base: Let $i = 0$ for the 0^{th} iteration.

$left = 0$, $l \geq 0$, $j \geq 0$, $right = n-1$, $n \geq 1$.

So $0 \leq 0 \leq 0 \leq 0 \leq 0$ ✓

Therefore the base case holds true.

Inductive Hypothesis: Assume $left_k \leq l_k \leq j_k \leq right_k$ and there exists a T.

Prove that $left_{k+1} \leq l_{k+1} \leq j_{k+1} \leq right_{k+1}$.

Also assume that $left_{k+1} \leq right_{k+1}$ from the last proof.

Inductive Step: Assuming the array contains distinct integers all > 0, we have 3 cases to look at assuming T exists: $x = T$, $x < T$, and $x > T$.

4. (Cont.)

Case 1: ($x = T$) Assume $left_k \leq l_k \leq j_k \leq right_k$. So the next iter., assuming the elements in the array are distinct, would be $l \leq j$. Left at most would equal $k+1$ and right would also be at least $k+1$. l and j would be unchanged on the next iteration so

$left_{k+1} \leq l_{k+1} \leq j_{k+1} \leq right_{k+1}$.

Case 2: ($x < T$) By the I.H., $\exists T$ and $left_k \leq l_k \leq j_k \leq right_k$. So on the $k+1^{th}$ iter, left++ and a[l++] and since the elements are distinct, $l{+}{+} \leq j$ and $left_{k+1} \leq right_{k+1}$. Putting it together, $left_{k+1} \leq l_{k+1} \leq j_{k+1} \leq right_{k+1}$

Case 3: ($x > T$) By the I.H., $\exists T$ and $left_k \leq l_k \leq j_k \leq right_k$. So on the $k+1^{th}$ iter., right -- and j-- and since the elements are distinct which means $k+1 \leq l \leq j$ assuming its sorted. $left_{k+1}$ at most $= k+1$ and $right_{k+1}$ at least $= k+1$ so putting it together:

$$left_{k+1} \leq l_{k+1} \leq j_{k+1} \leq right_{k+1}$$

Therefore since all of the cases and base hold True, we can say $left \leq l \leq j \leq right$.