COM S 311

Homework 1

Recitation 5, 1-2pm, Marios Tsekitsidis

Christian Shinkle

January 24, 2018

1. Base Case: Note

$$F_1^2 = F_1 * F_1$$
$$1^2 = 1 * 1$$
$$1 = 1$$

Inductive Hypothesis: $\forall n \leq 1, F_1^2 + F_2^2 + F_3^2 + ... + F_n^2 = F_n * F_{n+1}$
Goal: Prove $F_1^2 + F_2^2 + F_3^2 + ... + F_n^2 + F_{n+1}^2 = F_{n+1} * F_{n+2}$
Inductive Case: By I.H., $F_1^2 + F_2^2 + F_3^2 + ... + F_n^2 + F_{n+1}^2 = F_n * F_{n+1} + F_{n+1}^2$
$= F_{n+1}(F_n + F_{n+1})$
$= F_{n+1} * F_{n+2}$ by definition of Fibonacci Numbers

2. (a) Base Case: An FBT T which is a single node, so
$n(T) \geq 2h(T) - 1$
$1 \geq 2 * 1 - 1$
$1 \geq 1$
Inductive Hypothesis: Let T be an FBT with $n(T) \geq 2h(T) - 1$.
Inductive Case: Let R be a tree made of an FBT T plus two nodes attached to any leaf of T. By definition of FBT, R is an FBT because all of the leaves of T remain leaves, expect for the leaf with the two new attached nodes. Those two nodes become leaves and the node they are attached to becomes an internal node and now has two children. Note that two cases can occur when adding these nodes: the nodes increase the height of R or they do not.

Case 1: The height of R is not increased. Therefore,
$n(R) = n(T) + 2$
$\geq (2h(T) - 1) + 2$ by the I.H.
$= 2h(T) + 1$
Note $n(R) \geq 2h(T) + 1 > 2h(T) - 1$, so the property holds.

Case 2: The height of R is increased. This means the height of the R was increased by 1 because both nodes were attached to the same leaf. Therefore, $h(R) - 1 = h(T)$. So,
$n(R) = n(T) + 2$
$\geq (2h(T) - 1) + 2$ by the I.H.
$= (2(h(R) - 1) - 1) + 2$
$= (2h(R) - 3) + 2$
$= 2h(R) - 1$
Note, $n(R) \geq 2h(R) - 1$, so the property holds.
Both cases hold, so the property holds for R.

(b) Base Case: An FBT T which is a single node is a leaf, so

$$i(T) = (n(T) - 1)/2 = (1 - 1)/2 = 0$$

Inductive Hypothesis: Let X and Y be two FBT's with $i(X) = (n(X) - 1)/2$ and $i(Y) = (n(Y) - 1)/2$.
Goal: For tree R, $i(R) = (n(R) - 1)/2$.

Inductive Case: Let R be a FBT with root r and left child FBT X and right child FBT Y. Then, $i(R) = i(X) + i(Y) + 1$ because R consists of all nodes of X and Y plus the root r. By I.H., $i(R) = (n(X) - 1)/2 + (n(Y) - 1)/2) + 1$. Then, $i(R)$
$= \frac{(n(X)-1+n(Y)-1}{2} + 1$
$= \frac{(n(X)+n(Y)-2}{2} + 1$
$= \frac{(n(R)-1)-2}{2} + 1$ by definition of FBT
$= (n(R) - 1)/2 - 1 + 1$
$= (n(R) - 1)/2$

3. Base Case: Note $x = a, m = n, y = 1$. Therefore,
$a^n = x_0^{m_0} * y_0$
$= a^n * 1$
Inductive Hypothesis: $\forall i, a^n = x_i^{m_i} * y_i$ where i is the number of iterations through the loop.
Inductive Case: At the beginning of the $i + 1th$ iteration through the loop, one of two cases can arise, m is even or odd.
Case 1: m is even. Then,
$x_{i+1} = x_i^2$, $m_{i+1} = m_i/2$, and $y_{i+1} = y_i$. So,
$x_{i+1}^{m_{i+1}} * y_{i+1}$
$= x_i^{2*m_i/2} * y_i$
$= x_i^{m_i} * y_i$
By I.H., $a^n = x_i^{m_i} * y_i$, therefore the property holds.
Case 2: m is odd. Then,
$y_{i+1} = x_i * y_i$, $x_{i+1} = x_i^2$, and $m_{i+1} = (m_i - 1)/2$. So,
$x_{i+1}^{m_{i+1}} * y_{i+1}$
$= x_i^{2*(m_i-1)/2} * x_i * y_i$
$= x_i^{m_i-1} * x_i * y_i$
$= x_i^{m_i} * y_i$
By I.H., $a^n = x_i^{m_i} * y_i$, therefore the property holds.
Both cases hold, so the property holds for the $i + 1th$ iteration.

4. Base Case: Let $i = 0$ and $n =$ the length of Array a. Note $left = 0$, $right = n - 1$. Then, $left \leq right$ for any array with length ≥ 1. If $\exists l, j$ s.t. $a[l] + a[j] == T$, let $x = a[left] + a[right]$ which is equal to $a[0] + a[n - 1]$.
If $x == T$, return true and terminate.
Otherwise, if $x < T$, then $l > 0$ because $a[0]$ is the smallest element in a and a larger integer is necessary to satisfy $x == T$. The next larger element will be a[1] because a is sorted and all elements in a are distinct.
If $x > T$, then $j < n - 1$ because $a[n - 1]$ is the largest element and a smaller integer is necessary to satisfy $x == T$. The next smaller element is $a[n - 2]$ because the array is sorted and all elements in a are distinct.
If $left > right$ (i.e. a is of length 1) and x didn't equal T, then l and j don't exist because all indices have been checked. Then, return false and terminate.
Therefore, $left \leq l \leq j \leq right$.
Inductive Hypothesis: $\forall i$, the start of the ith iteration holds $left \leq right$ and if $\exists l, j$ s.t. $a[l] + a[j] == T$, then $left \leq l \leq j \leq right$.
Goal: Show at the start of the $i + 1th$ iteration, $left \leq right$ and if $\exists l, j$ s.t. $a[l] + a[j] == T$, then $left \leq l \leq j \leq right$.
Inductive Case: Proving $left \leq right$:
At the start of the $i + 1th$ iteration, by I.H., $left \leq right$. If $a[left] + a[right] == T$, then the program returns true and terminates, so $left$ will remain less than or equal to $right$. Otherwise, if $x < T$ then $left$ will be incremented by 1. If $left$ becomes greater than $right$, then the program will return false and terminate before the beginning of the next iteration. If $x > T$ then right will be decremented by 1. If $right$ becomes less than $left$, then the program will return false and terminate before the beginning of

the next iteration. Therefore, at the end of the $i+1th$ iteration, $left \leq right$.
Proving $left \leq l \leq j \leq right$:
At the start of the $i+1th$ iteration, by I.H., $left \leq l \leq j \leq right$. Let $x = a[left] + a[right]$. Then three cases can occur: $x == T$, $x < T$, or $x > T$.
Case 1: If $x == T$, then return true and terminate. This implies $left = l$ and $right = j$. Therefore, at the end of the $i+1th$ iteration, $left \leq l \leq j \leq right$ holds.
Case 2: If $x < T$ then a larger integer is necessary to satisfy $x == T$. Because a is sorted and all elements are distinct, $a[left+1] + a[right]$ will be the next larger number. Note $left < left + 1 \leq l \leq j \leq right$. Therefore, at the end of the $i+1th$ iteration, $left + 1 \leq l \leq j \leq right$ holds.
Case 3: If $x > T$ then a smaller integer is necessary to satisfy $x == T$. Because a is sorted and all elements are distinct, $a[left] + a[right-1]$ will be the next smaller number. Note $left \leq l \leq j \leq right - 1 < right$. Therefore, at the end of the $i+1th$ iteration, $left \leq l \leq j \leq right - 1$ holds.
All cases hold, so both properties hold for the $i+1th$ iteration.