

HW1

Shane Drafahl

30 August, 2017

1. Use mathematical induction to prove the following statements

(a)

$$\forall_n \geq 1, \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$$

lemma: $\sum_{i=1}^n i = \frac{n^2}{2} + \frac{n}{2}$ where $n \in \mathbb{N}$

Base Case: $\sum_{i=1}^1 i = 1 = \frac{1^2}{2} + \frac{1}{2}$

Inductive Hypothesis: $\sum_{i=1}^n i = \frac{n^2}{2} + \frac{n}{2}$

Inductive Steps:

Using induction on n

$$\frac{(n+1)^2}{2} + \frac{n+1}{2} = \frac{n^2}{2} + \frac{3n}{2} + 1 = \frac{n^2}{2} + \frac{n}{2} + (n+1)$$

So therefore the lemma is true for all natural numbers n greater than 1.

With the lemma we can solve the proof easier

Base Case: For n = 1 would be $\sum_{i=1}^1 i^3 = 1^3 = 1$. $(\sum_{i=1}^1 i)^2 = 1$.

Inductive Hypothesis: From the base n is true for $\forall_n \geq 1$, $\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$

Inductive Step: Using lemma we can re-write this to $(\frac{n^2}{2} + \frac{n}{2})^2 = \sum_{i=1}^n i^3$

We will first add n+1, $(\frac{(n+1)^2}{2} + \frac{(n+1)}{2})^2 = \frac{n^4}{4} + \frac{3n^3}{2} + \frac{13n^2}{4} + 3n + 1$. Now we will do $(\frac{n^2}{2} + \frac{n}{2})^2 + (n+1)^3 = \frac{n^4}{4} + \frac{3n^3}{2} + \frac{13n^2}{4} + 3n + 1$. So therefore $\forall_n \geq 1$, $\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$ is true for all natural numbers for n greater than or equal to 1.

(b) $\forall_n \geq 4, 2^n < n!$

Basis: for $n = 4$ $2^4 = 16$ and $4! = 24$

Inductive Hypothesis: $\forall_n \geq 4, 2^n < n!$ is true for n .

Inductive Step: $2^{n+1} = 2^n * 2$. $(n + 1)! = n! * (n + 1)$ Since we know from the inductive hypothesis that $\forall_n \geq 4, 2^n < n!$ so for this to remain true the right side and the left would need to be multiplied by the same value or the right would need to be multiplied by a greater value. The left is multiplied by 2 and the right is multiplied by $(n + 1)$. We know that n has to be 4 or greater so we know $(n + 1) \geq 2$ so therefore $\forall_n \geq 4, 2^n < n!$ for all natural numbers n .

2. Refer to the definition of Full Binary Tree from the notes. For a Full Binary Tree T , we use $n(T)$, $h(T)$, $i(T)$ and $l(T)$ to refer to number of nodes, height, number of internal nodes (non-leaf nodes) and number of leaves respectively. Note that the height of a tree with single node is 1 (not zero). Using structural induction, prove the following:

(a) For every Full Binary Tree T , $n(T) \geq h(T)$.

Basis: For a full binary tree T with a root node and two child leaf nodes $n(T_*) = 3$, and $h(T_*) = 2$ so $n(T_*) \geq h(T_*)$ is true.

Inductive Hypothesis: Suppose that for complete binary trees T , $n(T) \geq h(T)$ is true.

Recursive: Suppose we have two sub trees T_1, T_2 , these trees are identical full binary trees and when we combine them together with an additional node $T_1 + T_2 + 1 = T$, $n(T_{1,2}) \geq h(T_{1,2})$. First we will compare $n(T_1) + n(T_2) + 1 \geq h(T_{1,2}) + 1$. $n(T_1) + n(T_2) + 1 = n(T)$ and $h(T) = h(T_{1,2}) + 1$. We can reduce it down to $2n(T_1) + 1 \geq h(T_1) + 1 \dots 2n(T_1) \geq h(T_1)$. By the IH we know that $n(T_1) \geq h(T_1)$ is true so $2n(T_1) \geq h(T_1)$ must also be true so therefore $n(T) \geq h(T)$ is true for all full binary trees T .

(b) For every Full Binary Tree T , $i(T) \geq h(T) - 1$

Basis: For a full binary tree T_* with a single root node and two child leaf nodes. $i(T_*) = 1$ and $h(T_*) - 1 = 1$ so therefore $i(T_*) \geq h(T_*) - 1$

Inductive Hypothesis: Assume that $i(T) \geq h(T) - 1$ is true.

Recursive: We will take two identical full binary trees $T_{1,2}$. If we combined them with a single node to connect them into a new full binary tree T then to find all the internal nodes of T we would do $i(T_1) + i(T_2) + 1 \geq h(T) - 1$. The height of T is one more than T_1 so $i(T_1) + i(T_2) + 1 \geq h(T_1)$. This can also be reduced down to $i(2T_1) + 1 \geq h(T_1) - 1$. By the IH $i(T_1) + 1 \geq h(T_1) - 1$ so $i(2T_1) + 1 \geq h(T_1) - 1$ is also true and therefore $i(T) \geq h(T) - 1$ is true for all complete binary trees T .

(c) For every Full binary Tree T , $l(T) = (n(T) + 1)/2$

Basis: for a full binary tree T^2 with a single root node and two children has 2 leaf nodes and 3 altogether. so $(3 + 1)/2 = 2$, so therefore $l(T^3) = (n(T^3) + 1)/2$ is true for the basis.

Inductive Hypothesis: We assume $l(T) = (n(T) + 1)/2$ is true.

Recursive Step: We will take two identical full binary trees $T_{1,2}$. If we combined them with a single node to connect them into a new full binary tree T . By the IH $l(T_{1,2}^3) = (n(T_{1,2}^3) + 1)/2$. To find the number of leaves for T we can add these together so $\frac{n(T_1)+1}{2} + \frac{n(T_2)+1}{2}$. We can simplify this to because $T_1 = T_2$, $n(T_1) + 1 = l(T)$. Now we must prove this is correct. We know that $l(T) = (n(T) + 1)/2$ so we will expand $n(T)$ in terms of T_1 so $l(T) = (2n(T_1) + 1 + 1)/2 \dots l(T) = n(T_1) + 1$. So because we were able to prove for T with two sub trees therefore $l(T) = (n(T) + 1)/2$ is true for all full binary trees T .

3. Let a be an array of size n indexed by $0, 1, \dots, n-1$. Consider the following code (inner loop of Bubble Sort)

```
for i in the range [0, n-2]
if (a[i] > a[i+1])
swap (a[i], a[i+1]); //swap the values of a[i] and a[i+1]
```

Show the following using induction : At the start of i th iteration of the loop, $a[i]$ is the maximum among $a[0], a[1], \dots, a[i]$.

Basis: for $a[0]$ there are no values index in a below 0 so therefore $a[i]$ where $i = 0$ the start of the iteration of i is the maximum.

inductive Hypothesis: Assume that $a[i]$ is the greatest max value during the iteration and that $m(a[i])$ is either true or false if $a[i]$ is greatest during the i th iteration.

Inductive Step: By the IH assume that $m(a[i])$ is true so. If during the i th iteration $a[i] > a[i+1]$ then $a[i+1] = a[i]$ in that case for $m(a[i+1])$ $a[i+1] > a[i]$ and $m(a[i])$ is true by the inductive hypothesis so therefore for that that $m(a[i+1])$ is true. The other case is that $a[i+1] > a[i]$ during the i th iteration. In that case its the same where $m(a[i])$ is already true so starting at the $i+1$ iteration we already know $a[i]$ is greater than anything before it and the value after $a[i]$, is greater $a[i+1] > a[i]$ so therefore $m(a[i])$ or that $a[i]$ is the greatest max for $a[0], a[1], \dots, a[i]$ for all natural numbers i where $0 \leq i \leq n - 1$.

4. Let a be an array of size n indexed by $0, 1, \dots, n - 1$. Now consider Bubble sort pseudo code.

for j in the range $[0, n-1]$

for i in the range $[0, n-j-2]$ if $a[i] < a[i+1]$

swap($a[i], a[i+1]$)

Use mathematical induction to show: At the start of j th iteration of outer loop the following conditions hold:

$$a[n - j] \leq a[n - j + 1] \leq a[n - 1]$$

$a[n - j], a[n - j + 1], \dots, a[n - 1]$ are the j largest elements of the array

Basis: Using induction on j if $j = 2$ then $a[n - 2] \leq a[n - 1] \leq a[n - 1]$ obviously $a[n - 1] \leq a[n - 1]$. For arrays greater than 1 element. For the prior iteration where $j = 1$ then the inner for loop would have went from 0 to $n-1$. For the inner loop when $i = n - 2$ if $a[n - 2] > a[n - 1]$ then it would swap and put the greater value in $a[n - 1]$ otherwise the greater value would already be there. So $a[n - 2] < a[n - 1]$.

Inductive Hypothesis: Assume that $a[n - j] \leq a[n - j + 1] \leq a[n - 1]$ is true.

Inductive Step: Now we must prove for $j + 1$ so $a[n - j + 1] \leq a[n - j + 2] \leq a[n - 1]$. We know that j works so during the j th iteration