

Problem Set 9

David Bungler

December 2022

Problem 1

Theorem:

Starting at a positive integer n , the sequence created by applying the function $f(n)$ to the previous output of the function will reach 1 after at most $3\lceil\log_2 n\rceil$ applications of f .

Proof:

Let $A(x)$ be the number of applications required for a sequence starting at x to reach 1.

Base Cases:

$n = 1$: Since n is already at 1, $A(1) = 0$. $3\lceil\log_2(1)\rceil = 3(0) = 0$. $0 \leq 0$, so the theorem holds.

$n = 2$: Since n is even, one application of $f(n)$ would result in 1, since $2/2 = 1$, meaning $A(2) = 1$. $3\lceil\log_2(2)\rceil = 3(1) = 3$. Since $1 \leq 3$, the theorem holds.

Inductive cases:

Assume that $A(j) \leq 3\lceil\log_2 k\rceil$ is true for all positive integers j from 1 to k , where $k \geq 2$. We are trying to prove that $A(k+1) \leq 3\lceil\log_2(k+1)\rceil$

Case 1: $k+1$ is even

After one application of the function, $k+1$ would become $\frac{k+1}{2}$, meaning $A(k+1) = A(\frac{k+1}{2}) + 1$. Because $\frac{k+1}{2}$ is less than k , by the Inductive Hypothesis we know that $A(\frac{k+1}{2}) \leq 3\lceil\log_2(\frac{k+1}{2})\rceil$, which can be rewritten as $3\lceil\log_2(k+1) - 1\rceil$. This then becomes $3\lceil\log_2(k+1)\rceil - 3$. By adding 1 to both sides, we get $A(\frac{k+1}{2}) + 1 \leq 3\lceil\log_2(k+1)\rceil - 2$. Since $A(k+1) = A(\frac{k+1}{2}) + 1$ and since $3\lceil\log_2(k+1)\rceil - 2 < 3\lceil\log_2(k+1)\rceil$, we get $A(k+1) \leq 3\lceil\log_2(k+1)\rceil$.

Case 2: $k + 1$ is odd

After one application of the function, $k + 1$ would become $2(k + 2)$. After another application of the function, it would become $k + 2$, and then after a third application, it would become $\frac{k+2}{2}$. This means that $A(k + 1) =$

$A(\frac{k+2}{2}) + 3$ By the inductive hypothesis, we know that $A(\frac{k+2}{2}) \leq 3\lceil\log_2(\frac{k+2}{2})\rceil$. $3\lceil\log_2(\frac{k+2}{2})\rceil$ can be rewritten as $3\lceil\log_2(k + 2) - 1\rceil$. From there, we get $A(\frac{k+2}{2}) \leq 3\lceil\log_2(k + 2)\rceil - 3$. By adding 3 to both sides, we get $A(\frac{k+2}{2}) + 3 \leq 3\lceil\log_2(k + 2)\rceil$.

Because k is an integer, the only way for $\lceil\log_2(k + 1)\rceil$ to be greater than $\lceil\log_2(k + 2)\rceil$ is if $\log_2(k + 1)$ is a whole number, meaning $\log_2(k + 2)$ would be slightly greater and then rounded up. If $\log_2(k + 2)$ were a whole number, both $\lceil\log_2(k + 2)\rceil$ and $\lceil\log_2(k + 1)\rceil$ would be rounded up to $\log_2(k + 2)$. In all other cases, $\log_2(k + 1)$ and $\log_2(k + 1)$ are between the same two whole number by the properties of logarithms. However, since all of the powers of 2 are even, $\log_2(k + 1)$ cannot be a whole number since $k + 1$ is odd. This means that $\lceil\log_2(k + 2)\rceil$ is always equal to $\lceil\log_2(k + 1)\rceil$ in this case.

Using this, we can say that $A(\frac{k+2}{2}) + 3 \leq 3\lceil\log_2(k + 1)\rceil$, which means that $A(k + 1) \leq 3\lceil\log_2(k + 1)\rceil$.

Since Cases 1 and 2 cover all possible values for $k + 1$ where $k \geq 2$, this proves the theorem.

Problem 2

Theorem:

Given a country of n cities in Minecraft, where $n \geq 2$, that are all connected to each other city by one or more roads, there will always be at least two different cities that can be removed individually while still maintaining the connectivity between the remaining cities.

Proof:

Base Case:

$n = 2$: Removing either city would leave one remaining city, meaning that the one remaining city is connected to every city.

Inductive Step:

Case 1: There exists a city whose removal disconnects the network

Assume that all countries with a number of cities j from 2 to k , where $k \geq ?$, contain at least 2 cities that, when removed, do not disconnect any cities in the network. This means that each of these countries has at most $j - 2$ cities that, when removed, disconnect the network. We must show that this also holds for $k + 1$, meaning a country with $k + 1$ cities has at most $k - 1$ cities that, when removed, disconnect the network. A city is irremovable when it is the only point that connects two distinct sections of the network together.

Sub Case 1: Adding a city with only one connection When adding a city to a network, if it is only added with one connection, that new city becomes a distinct section of the network, and the city it connects to becomes irremovable. This would decrease the number of removable cities by one. However, as the new city is only connected to one other city, removing it wouldn't disconnect anything, meaning that city is itself a removable city. Since one removable city was gained and one was lost, the net change in the number of removable cities is zero. By the Inductive Hypothesis, the original network had at least 2 removable cities, meaning the new network also has at least 2 removable cities, proving the theorem.

Sub Case 2: Adding a city with multiple connections

In the case that the new city is connected to multiple cities, it does not create any distinct sections, since the other cities that the new one is connected to had all been previously connected before the new one was added. Regardless of if the new city is removable or not, its addition didn't create any new irremovable cities. Again, by the Inductive Hypothesis, the original network had at least two removable cities, and since none became irremovable, the new network also has at least two removable cities.

Case 2: There are no cities whose removal disconnects the network

In this case, any city can be removed without disconnecting the network. Since there are at least 2 cities, that mean there always exists at least 2 cities that can be removed while maintaining connectivity.

Problem 3

Theorem:

For any root tree x , if x has m nodes, then it has $m - 1$ edges.

Base Case:

$m = 1$: A root tree with only one node means there are no connections to any other nodes, and therefore there are 0, or $m - 1$ edges.

Inductive Case:

Assume that a root tree x has n_x nodes and $n_x - 1$ edges. We are trying to prove that the theorem maintains for a tree with $n_x + 1$ nodes. By adding one node to x , it also requires an extra edge to connect it into the tree. This means that this new larger tree would have $n_x + 1$ nodes and n_x edges, maintaining the theorem. This process can be done at any section of the tree, as long as it maintains the properties of a root tree.

Problem 4

0.

$$c = \lceil \log_3(k) \rceil \wedge k < 3x$$

1.

$$k = 1, c = 0, x \geq 1$$

$$0 = \lceil \log_3(1) \rceil$$

$$3x \geq 3$$

$$1 < 3x$$

2.

Before:

$$c_1 = \lceil \log_3(k_1) \rceil \wedge k_1 < 3x$$

After:

$$k_2 = 3k_1, c_2 = c_1 + 1$$

$$c_1 = \lceil \log_3(k_1) \rceil$$

$$c_1 + 1 = \lceil \log_3(k_1) \rceil + \lceil \log_3(3) \rceil$$

$$\text{by adding one to each side.} = c_1 + 1 = \lceil \log_3(3k_1) \rceil$$

$$= c_2 = \lceil \log_3(k_2) \rceil$$

$$k_2 < x \text{ by the loop condition, therefore, } k_2 < 3x$$

3.

Skipped

4.

Prove that if $\lceil \log_3(k) \rceil = c \wedge x \leq k < 3x$, then $\lceil \log_3(x) \rceil = c$

Assume $c = \lceil \log_3(k) \rceil \wedge x \leq k < 3x$.

Since $\lceil \log_3(k) \rceil = c$, then $\log_3(k) \leq c < \log_3(k) + 1$.

If we take \log_3 of all of the parts of $x \leq k < 3x$, we get

$$\log_3(x) \leq \log_3(k) < \log_3(3x)$$

We know this is still true because \log_3 is an increasing function, and because x and k are positive numbers due to the preconditions of the function. We also know that $\log_3(3x) = \log_3(x) + 1$, giving us

$$\log_3(x) \leq \log_3(k) < \log_3(x) + 1$$

Since $\log_3(x) \leq \log_3(k)$ and $\log_3(k) \leq c$, we know that $\log_3(x) \leq c$ by the transitive property of inequalities.

Since $x \leq k$, that means $3x \leq 3k$, which also means $\log_3(3x) \leq \log_3(3k)$, or $\log_3(x) + 1 \leq \log_3(k) + 1$.

Because of this, because $\log_3(k) < \log_3(x) + 1$, and because

$\log_3(k) \leq c < \log_3(k) + 1$, this means that $c < \log_3(x) + 1$.

Because we've proven that $\log_3(x) \leq c < \log_3(x) + 1$, we know that $\lceil \log_3(x) \rceil = c$.