Problem 1.
Consider the sequence $a_1 = 1$, $a_2 = 2$, $a_3 = 3$, and, for $n > 3$,

$$a_n = a_{n-1} + a_{n-2} + a_{n-3}$$

Prove that $a_n < 2^n$ for $n \geq 1$. What kind of induction did you use?

Base Cases

Suppose $n = 1$	Suppose $n = 2$	Suppose $n = 3$
$a_n = 1 < 2 = 2^1$	$a_n = 2 < 4 = 2^2$	$a_n = 3 < 8 = 2^3$

Inductive Hypothesis
Assume that

$$\forall h \;\; h \leq k \qquad a_h < 2^h$$

Inductive Step
Must prove $a_{k+1} < 2^{k+1}$

$a_{k+1} = a_k + a_{k-1} + a_{k-2}$

$< 2^k + 2^{k-1} + 2^{k-2}$ Inductive Hypothesis

$< 2^k + 2^{k-1} + 2^{k-1}$

$= 2^k + 2(2^{k-1})$

$= 2^k + 2^k$

$= 2(2^k)$

$= 2^{k+1}$

So if the statement is true from $(k - 2)$ up to k, it's true for $k+1$.
The statement is true for 1, 2, and 3. Thus, it's true for all integers afterward.

Thus, we've proven it by Strong Induction.

Done.

Problem 2.
Let $x_1 = 1$ and $x_{n+1} = \sqrt{1+2x_n}$ for $n \geq 1$. Prove that $x_n < 4$ for $n \geq 1$.

Base Case
Suppose $n = 1$

$x_n = x_1 = 1 < 4$

Inductive Hypothesis
Assume for $k \geq 1$

$$x_k < 4$$

Inductive Step

$x_{k+1} = \sqrt{1+2x_k}$

$< \sqrt{1+2(4)}$ Inductive Hypothesis

$< \sqrt{9} = 3 < 4$

Done.

Problem 3.
Let $f(n) = f(n/2) + 4n$ for $n = 2^k$ and $k \geq 1$; $f(1) = 1$. Prove that $f(n) \leq 8n$.

Clearly, it's true for $n = 1$ since $f(1) = 1 \leq 8(1) = 8n$

Notice that the function's domain is consists of powers of 2. So instead do our induction on k, as this will go through the rest of the domain.

<u>Base Case</u>
Suppose $k = 1$. Then $n = 2^1 = 2$

$$f(n) = f(n/2) + 4n = f(2/2) + 4(2) = f(1) + 8 = 1 + 8 = 9 \leq 16 = 8(2) = 8n$$

<u>Inductive Hypothesis</u>
Assume for $l = 2^a$ for $a \geq 1$

$$f(l) \leq 8l \quad \text{(or, equivalently, } f(2^a) \leq 8(2^a))$$

<u>Inductive Step</u>
Must show $f(2l) \leq 8(2l)$.

$f(2l) \leq 8(2l)$

$f(2(2^a)) \leq 8(2(2^a))$ $\qquad$ $l = 2^a$

$f(2^{a+1}) \leq 8(2^{a+1})$

So we'll prove that

$f(2^{a+1}) = f(2^{a+1} / 2) + 4(2^{a+1})$

$f(2^a) + 4(2^{a+1})$

$\leq 8(2^a) + 4(2^{a+1})$ $\qquad$ Inductive Hypothesis

$= 4(2^{a+1}) + 4(2^{a+1})$

$= 8(2^{a+1})$

So if the statement is true for l, it's true for $2l$. The statement is true for 2.
Thus, it's true for all powers of 2 thereafter.

Done.

Problem 4.
Which of the following recursive functions are well-defined for integer $n \geq 0$:

$$f(n) = \begin{cases} 1 & ,n = 0 \\ 2f(n-1) & ,n > 0 \end{cases}$$

$$f(n) = \begin{cases} 1 & ,n = 0 \\ f(n+1) - 1 & ,n > 0 \end{cases}$$

$$f(n) = \begin{cases} 1 & ,n = 0 \\ nf(n-1) & ,n > 0 \end{cases}$$

$$f(n) = \begin{cases} 1 & ,n = 0 \\ f(n-2) + 2 & ,n > 0 \end{cases}$$

A recursive function is well defined if there is a specific output for any given input in the domain.

The first function is one such function. For f(0), we have the value 1, and for any other *n*, we may recursively apply the formula all the way back to 1.

$$f(n) = 2\ f(n-1) = 4\ f(n-2) = \ldots = 2^{n-1}\ f(1) = 2^n\ f(0) = 2^n$$

The second function is not well defined. Notice that each input depends on the *next*.

What is f(n)? Oh it's based on f(n+1). But what's that? Something using f(n+2) ….

The third function is well defined, since each value depends on the previous, and the first case is given.

It's also just the factorial function, f(n) = n!, which we've proven before.

The fourth function is not well defined.

For $2 \mid n$ we have

f(n) depends on f(n – 2)
which depends on f(n – 4)
…
which depends on f(2)
which depends on f(0), a value we know

For $2 \nmid n$ we have

f(n) depends on f(n – 2)
which depends on f(n – 4)
…
which depends on f(1)
which depends on f(-1), a value that's undefined.

Problem 1.

The three recursive functions appear only slightly different. In each case guess a non-recursive formula for f(*n*) and prove your guess by induction.

- $f(0) = 0$; $f(n) = 2 + f(n-1)$, for integer $n > 0$.
- $f(0) = 0$; $f(n) = 2f(n-1)$, for integer $n > 0$.
- $f(0) = 1$; $f(n) = 2f(n-1)$, for integer $n > 0$.

We observe 0, 2, 4, 6, … 2n. Base Case $n = 1$ $2n = 2(1) = 2$ $f(1) = 2 + f(0) = 2 + 0 = 2$ Inductive Hypothesis Assume for $k \geq 1$ $f(k-1) = 2(k-1)$ Inductive Step $f(k) = 2 + f(k-1)$ $2 + 2(k-1)$ I.H. $2 + 2k - 2$ $2k$	We observe 0, 0, 0, 0, … 0. Base Case $n = 1$ 0 $f(1) = 2\ f(0) = 2(0) = 0$ Inductive Hypothesis Assume for $k \geq 1$ $f(k-1) = 0$ Inductive Step $f(k) = 2\ f(k-1)$ $2(0)$ I.H. 0	We observe 1, 2, 4, 8, … 2^n. Base Case $n = 1$ $2^n = 2^1$ $f(1) = 2\ f(0) = 2(1) = 2^1$ Inductive Hypothesis Assume for $k \geq 1$ $f(k-1) = 2^{k-1}$ Inductive Step $f(k) = 2\ f(k-1)$ $2(2^{k-1})$ I.H. 2^k

Problem 2.

Consider the following pseudocode:

```
1: Function Big(n)
2: if n = 0 then
3:     return(1)
4: else
5:     return(2 × Big(n − 1))
6: end if
```

Prove by induction that the output of Big(n) is 2^n.
Prove by induction that the running time of Big(n) is O(n).

First, we'll prove the output is 2^n.

<u>Base Case</u>
Suppose $n = 0$

$$\text{Big}(n) = \text{Big}(0) = 1 = 2^0$$

<u>Inductive Hypothesis</u>
Assume for $k \geq 0$

$$\text{Big}(k) = 2^k$$

<u>Inductive Step</u>
Must prove the statement for k+1

Big(k + 1)	
2 Big(k + 1 – 1)	Line 5 says we double the prior
2 Big(k)	
$2\ (2^k)$	By the Inductive Hypothesis
2^{k+1}	

So it's true by induction.

```
Function Big(n)
if n = 0 then
    return(1)
else
    return(2 × Big(n − 1))
end if
```

Now, we'll prove the complexity is O(*n*).

Let *R*(*n*) be the running time for a given input *n*.

We'll say R(0) ≤ *d*, for some constant amount of work *d*.

Here, we are saying *d* is the maximal cost of computing one frame of the function.

So, in total, R(*n*) ≤ *d* + R(*n* – 1) since:

- Computing the initial frame is *d*.
- More work is done after the recursive call, which equals R(n – 1).

To show the algorithm is O(*n*), we'll prove that, for some constant *c* and cutoff *k*,

$$\forall n \;\; n \geq k \rightarrow R(n) \leq cn$$

We can pick whatever *c* we want, so we'll choose *c* ≥ 2*d*.

<u>Base Case</u>

Notice, the claim isn't true for *n* = 0.

R(0) ≤ d   but *d* > cn = c(0) = 0.

Suppose *n* = 1

R(1) ≤ d + R(0) ≤ d + d = 2d ≤ c = cn   So we'll let our cutoff, *k*, be 1.

<u>Inductive Hypothesis</u>

Assume for *h* ≥ 1

$$R(h) \leq ch$$

<u>Inductive Step</u>

R(h + 1) ≤ d + R(h)

≤ d + ch   Inductive Hypothesis
≤ c + ch   since we've chosen c ≥ 2d.
≤ c(h + 1)

Done.