Adam Blank Spring 2017

CSE 311

Foundations of Computing I

Euclid's Algorithm

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

```
gcd(a, b) {
   if (b == 0) {
      return a;
   }
   else {
      return gcd(b, a mod b);
   }
}
```

Finding x & y

Bézout's Theorem

If a and b are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that

$$\gcd(a, b) = ax_{(a,b)} + by_{(a,b)}$$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

Finding x & y

Bézout's Theorem

If a and b are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that $\gcd(a, b) = ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

Case 1: `gcd(a, 0) = a`

$\gcd(a, 0) = a*X_{a,0} + 0*Y_{a,0}$

Finding x & y

Bézout's Theorem

If a and b are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that $\gcd(a, b) = ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

We've figured out the answer for the "base case".

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$\gcd(a, b) = aX_{a,b} + bY_{a,b}$

Finding x & y

Bézout's Theorem

If a and b are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that $\gcd(a, b) = ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$\gcd(a, b) = aX_{a,b} + bY_{a,b}$

$= \gcd(b, a \bmod b) = ?????????$

We're stuck. We need to find $X_{a,b}$ and $Y_{a,b}$.

We're looking for an equation with a*x + b*y. The "a mod b" doesn't belong.

$\gcd(b, a \bmod b) = bX_{b,a \bmod b} + (a \bmod b)Y_{b,a \bmod b}$

Division Theorem

$a = b(a \operatorname{div} b) + (a \bmod b)$

Finding x & y

Bézout's Theorem
If *a* and *b* are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that gcd(a, b) = $ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\gcd(a, b) = aX_{a,b} + bY_{a,b}$$
$$= \gcd(b, a \bmod b) = ?????????$$

We're stuck. We need to find $X_{a,b}$ and $Y_{a,b}$.
We're looking for an equation with a*x + b*y. The "a mod b" doesn't belong.

$$\gcd(b, a \bmod b) = bX_{b,a \bmod b} + (a \bmod b)Y_{b,a \bmod b}$$
$$= bX_{b,a \bmod b} + (a - b(a \text{ div } b))Y_{b,a \bmod b}$$

Division Theorem

$$a = b(a \text{ div } b) + (a \bmod b)$$
$$(a \bmod b) = a - b(a \text{ div } b)$$

Finding x & y

Bézout's Theorem
If *a* and *b* are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that gcd(a, b) = $ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\gcd(a, b) = aX_{a,b} + bY_{a,b}$$
$$= \gcd(b, a \bmod b) = ?????????$$

$$\gcd(b, a \bmod b) = bX_{b,a \bmod b} + (a \bmod b)Y_{b,a \bmod b}$$
$$= bX_{b,a \bmod b} + (a - b(a \text{ div } b))Y_{b,a \bmod b}$$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has x and y terms. Group them...

Finding x & y

Bézout's Theorem
If *a* and *b* are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that gcd(a, b) = $ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\gcd(a, b) = aX_{a,b} + bY_{a,b}$$
$$= \gcd(b, a \bmod b) = ?????????$$

$$\gcd(b, a \bmod b) = bX_{b,a \bmod b} + (a \bmod b)Y_{b,a \bmod b}$$
$$= bX_{b,a \bmod b} + (a - b(a \text{ div } b))Y_{b,a \bmod b}$$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has x and y terms. Group them...

$$= bX_{b,a \bmod b} + aY_{b,a \bmod b} - b(a \text{ div } b)Y_{b,a \bmod b}$$
$$= b(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b}) + aY_{b,a \bmod b}$$
$$= aY_{b,a \bmod b} + b(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b})$$

$$\gcd(b, a \bmod b) = aY_{b,a \bmod b} + b(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b})$$

Finding x & y

Bézout's Theorem
If *a* and *b* are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that gcd(a, b) = $ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\gcd(a, b) = aX_{a,b} + bY_{a,b}$$
$$= \gcd(b, a \bmod b)$$
$$= aY_{b,a \bmod b} + b(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b})$$

Finding x & y

Bézout's Theorem
If *a* and *b* are positive integers, then there exist integers $x_{(a,b)}$ and $y_{(a,b)}$ such that gcd(a, b) = $ax_{(a,b)} + by_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\gcd(a, b) = aX_{a,b} + bY_{a,b}$$
$$= \gcd(b, a \bmod b)$$
$$= aY_{b,a \bmod b} + b(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b})$$

EGCD Algorithm

```
egcd(a, 0) = a*1 + 0*0
egcd(a, b) = a*Y_{b,a mod b} + b*(X_{b,a mod b} - (a div b)Y_{b,a mod b})
```

Finding x & y

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

EGCD Algorithm

```
egcd(a, 0) = a*1 + 0*0
egcd(a, b) = a*Y_{b,a mod b} + b*(X_{b,a mod b} - (a div b)Y_{b,a mod b})
```

$\gcd(a,b) = ax + by$

EGCD Algorithm

```
egcd(a, 0) = (a, 1, 0)
egcd(a, b) = (gcd(b, a mod b), Y_{b,a mod b}, X_{b,a mod b} - (a div b)*Y_{b,a mod b})
```

$\text{egcd}(b, a \bmod b) = (\gcd(b, a \bmod b), X_{b,a \bmod b}, Y_{b,a \bmod b})$

Large Non-negative Integer Operations MOD M

Division?

FINALLY! We're back to division mod m.

In normal arithmetic, if I multiply x * (1/x), I get back 1.

In MODULAR arithmetic, if I multiply x * ?, I get back 1.

"1/x" is the unique number that, when multiplied by x gives 1.

Large Non-negative Integer Operations MOD M

Division?

FINALLY! We're back to division mod m.

In normal arithmetic, if I multiply x * (1/x), I get back 1.

In MODULAR arithmetic, if I multiply x * ?, I get back 1.

"1/x" is the unique number that, when multiplied by x gives 1.

"1/x" is a solution, N, to the equation $xN \equiv 1 \pmod{m}$.

$$\begin{aligned} xN \equiv 1 \pmod{m} &\leftrightarrow m \mid (xN - 1) \\ &\leftrightarrow xN - 1 = km \\ &\leftrightarrow xN + (-k)m = 1 \end{aligned}$$

We know how to do this now! It's just EGCD!

Administrivia

If you want to use a token on HW1-HW3, you need to sign up for it by 11:30pm tonight.

Midterm practice materials are up on the website.

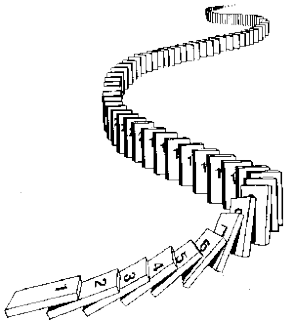
The midterm will be on Wed, May 3 from 4:00pm – 5:30pm in KNE 120.

If you cannot make this time, I need to know by **Friday** to schedule a make-up exam.

There will be two review sessions time/location TBD.

CSE 311: Foundations of Computing

Lecture 14: Induction



Mathematical Induction

Method for proving statements about all natural numbers

- A new logical inference rule!
  - It only applies over the natural numbers
  - The idea is to **use** the special structure of the naturals to prove things more easily
- Particularly useful for reasoning about programs!

```
for(int i=0; i < n; n++) { … }
```

  - Show P(i) holds after i times through the loop

```
public int f(int x) {
    if (x == 0) { return 0; }
    else { return f(x - 1); }
}
```

  - f(x) = x for all values of x ≥ 0 naturally shown by induction.

Prove $\forall(a, b \in \mathbb{Z})\ \forall(n \in \mathbb{N})\ (a \equiv b \pmod{n} \rightarrow a^i \equiv b^i \pmod{n})$

Let $a, b \in \mathbb{Z}$ be arbitrary. Let $i \in \mathbb{N}$ be arbitrary.

Suppose $a \equiv b \pmod{n}$.

We know $(a \equiv b \pmod{n} \wedge a \equiv b \pmod{n}) \rightarrow a^2 \equiv b^2 \pmod{n}$ by multiplying congruences. So, applying this repeatedly, we have:

$$(a \equiv b \pmod{n} \wedge a \equiv b \pmod{n}) \rightarrow a^2 \equiv b^2 \pmod{n}$$
$$(a^2 \equiv b^2 \pmod{n} \wedge a \equiv b \pmod{n}) \rightarrow a^3 \equiv b^3 \pmod{n}$$
$$\ldots$$
$$(a^{i-1} \equiv b^{i-1} \pmod{n} \wedge a \equiv b \pmod{n}) \rightarrow a^i \equiv b^i \pmod{n}$$

The "…"s is a problem! We don't have a proof rule that allows us to say "do this over and over".

So, make one!

Domain: Natural Numbers

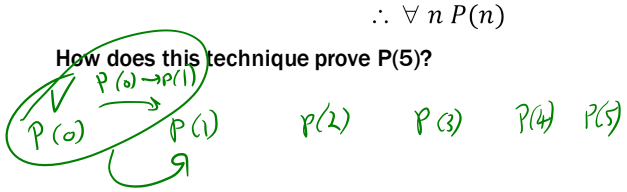
$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

Induction Is A Rule of Inference

Domain: Natural Numbers

$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

How does this technique prove P(5)?

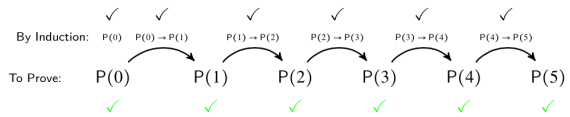


Induction Is A Rule of Inference

Domain: Natural Numbers

$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

How does this technique prove P(5)?

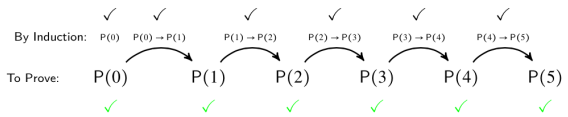


Induction Is A Rule of Inference

Domain: Natural Numbers

$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

How does this technique prove P(5)?



First, we prove P(0).
Since P(n) → P(n+1) for all n, we have P(0) → P(1).
Since P(0) is true and P(0) → P(1), by Modus Ponens, P(1) is true.
Since P(n) → P(n+1) for all n, we have P(1) → P(2).
Since P(1) is true and P(1) → P(2), by Modus Ponens, P(2) is true.

Using The Induction Rule In A Formal Proof

$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

1. Prove P(0)
2. Let k be an arbitrary integer ≥ 0
   - 3.1. Assume that P(k) is true
   - 3.2. …
   - 3.3. Prove P(k+1) is true
3. P(k) → P(k+1) — Direct Proof Rule
4. ∀k (P(k) → P(k+1)) — Intro ∀: 2, 3
5. ∀n P(n) — Induction: 1, 4

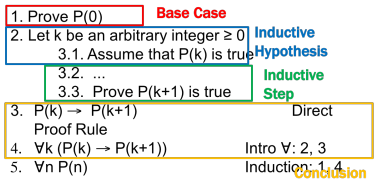
Translating to an English Proof

$$P(0)$$
$$\forall\, k\ (P(k) \rightarrow P(k+1))$$
$$\therefore\ \forall\, n\ P(n)$$

1. Prove P(0) — **Base Case**
2. Let k be an arbitrary integer ≥ 0 — **Inductive Hypothesis**
   - 3.1. Assume that P(k) is true
   - 3.2. …
   - 3.3. Prove P(k+1) is true — **Inductive Step**
3. P(k) → P(k+1) — Direct Proof Rule
4. ∀k (P(k) → P(k+1)) — Intro ∀: 2, 3
5. ∀n P(n) — Induction: 1, 4

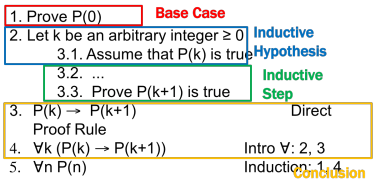
Conclusion

Translating To An English Proof



We will show that P(n) is true for every $n \in \mathbb{N}$ by Induction.

Translating To An English Proof



Induction Proof Template

[...Define P(n)...]
We will show that P(n) is true for every $n \in \mathbb{N}$ by Induction.
Base Case: *[...proof of P(0) here...]*
Induction Hypothesis:
Suppose P(k) is true for some $k \in \mathbb{N}$.
Induction Step:
We want to prove that P(k+1) is true.
[...proof of P(k+1) here...]
*The proof of P(k+1) **must** invoke the IH somewhere.*
So, the claim is true by induction.

5 Steps To Inductive Proofs In English

Proof:

1. "We will show that P(n) is true for every n ≥ 0 by Induction."
2. "Base Case:" Prove P(0)
3. "Inductive Hypothesis:"
   Assume P(k) is true for some arbitrary integer k ≥ 0"
4. "Inductive Step:" Want to prove that P(k+1) is true:
   Use the goal to figure out what you need.
   Make sure you are using I.H. and point out where you are using it. (Don't assume P(k+1) !!)
5. "Conclusion: Result follows by induction"

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

- We could try proving it with properties of summations?
- We could use calculus?
- Could this be induction?

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0):

Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

- We could try proving it with properties of summations?
- We could use calculus?
- Could this be induction?

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0):

Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Induction Hypothesis:

Suppose P(k) is true for some $k \in \mathbb{N}$.

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0):

Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Induction Hypothesis:

Suppose P(k) is true for some $k \in \mathbb{N}$.

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0):

Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Induction Hypothesis:

Suppose P(k) is true for some k ∈ ℕ.

Induction Step:

We want to show P($k+1$). That is, we want to show:

$$\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1$$

$\sum_{i=0}^{k+1} 2^i = \ldots = \ldots = \ldots = 2^{n+1} - 1.$

One of these steps must use the IH.

So, the claim is true for all natural numbers by induction.

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0): Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Induction Hypothesis: Suppose P(k) is true for some k ∈ ℕ.

Induction Step: We want to show P($k+1$). That is, we want to show: $\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1$

Note that $\sum_{i=0}^{k+1} 2^i =$

We know (by IH)...

$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$

We're trying to get...

This is exactly P($k+1$). So, P(k) → P($k+1$).

So, the claim is true for all natural numbers by induction.

Our goal is to find a sub-expression of the left that looks like the left side of the IH.

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

Let $P(n)$ be "$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$". We go by induction on n.

Base Case (n=0): Note that $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$, which is exactly $P(0)$.

Induction Hypothesis: Suppose P(k) is true for some k ∈ ℕ.

Induction Step: We want to show P($k+1$). That is, we want to show: $\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1$

Note that $\sum_{i=0}^{k+1} 2^i = \left(\sum_{i=0}^{k} 2^i\right) + 2^{k+1}$ [Splitting the summation]

$= (2^{k+1} - 1) + 2^{k+1}$ [By IH]

$= (2^{k+1} + 2^{k+1}) - 1$ [Assoc. of +]

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

$= 2^{k+2} - 1$ [Simplifying]

Don't bother justifying the "obvious" steps. But make sure you say "by IH" somewhere.

This is exactly P($k+1$). So, P(k) → P($k+1$).

So, the claim is true for all natural numbers by induction.

We know (by IH)...

$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$

We're trying to get...

$\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1$

Our goal is to find a sub-expression of the left that looks like the left side of the IH.

Prove $1 + 2 + 3 + \ldots + n = n(n+1)/2$

Let $P(n)$ be "$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$". We go by induction on n.

Base Case (n=0):

Induction Hypothesis:

Induction Step:

This is exactly P($k+1$). So, P(k) → P($k+1$).

So, the claim is true for all natural numbers by induction.

We know (by IH)...

We're trying to get...

Our goal is to find a sub-expression of the left that looks like the left side of the IH.

Prove $1 + 2 + 3 + \ldots + n = n(n+1)/2$

Let $P(n)$ be "$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$". We go by induction on n.

Base Case (n=0): Note that $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$, which is exactly $P(0)$.

Induction Hypothesis: Suppose P(k) is true for some k ∈ ℕ.

Induction Step: We want to show P($k+1$). That is, we want to show: $\sum_{i=0}^{k+1} i = \frac{(k+1)(k+2)}{2}$

Note that $\sum_{i=0}^{k+1} i = \left(\sum_{i=0}^{k} i\right) + (k+1)$ [Splitting the summation]

$= \left(\frac{k(k+1)}{2}\right) + (k+1)$ [By IH]

$= (k+1)\left(\frac{k}{2} + 1\right) = (k+1)\left(\frac{k+2}{2}\right)$ [Algebra]

$= \frac{(k+1)(k+2)}{2}$ [Algebra]

This is exactly P($k+1$). So, P(k) → P($k+1$).

So, the claim is true for all natural numbers by induction.

We know (by IH)...

$\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$

We're trying to get...

$\sum_{i=0}^{k+1} i = \frac{(k+1)(k+2)}{2}$

Our goal is to find a sub-expression of the left that looks like the left side of the IH.

Prove $3 \mid 2^{2n} - 1$ for all $n \geq 0$.

Let $P(n)$ be "$3 \mid 2^{2n-1}$." We go by induction on n.

Base Case (n=0):

Induction Hypothesis:

Induction Step:

We know (by IH)...

...which means...

We're trying to get...

...which is true if...

Prove $3 \mid 2^{2n} - 1$ for all $n \geq 0$.

Let $P(n)$ be "$3 \mid 2^{2n-1}$." We go by induction on n.

Base Case (n=0): Note that $2^{2 \cdot 0} - 1 = 2^0 - 1 = 1 - 1 = 0$.

We know $3 \mid 0$, by definition of divides, because $3 \cdot 0 = 0$. So, P(0) is true.

Induction Hypothesis: Suppose P(k) is true for some $k \in \mathbb{N}$.

Induction Step: We want to show P($k+1$). That is, WTS $3 \mid 2^{2(k+1)-1}$.

Note that $2^{2(k+1)-1} - 1 = 2^{2k+2-1} - 1$ [Algebra]

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

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

By IH, we know $3 \mid 2^{2k-1} - 1$. So, by definition of divides, we know $2^{2k-1} - 1 = 3j$ for some j.

$= (3j + 1)(4) - 1 = 3(4j + 1)$ [Algebra]

So, by definition of divides, $3 \mid 2^{2(k+1)-1} - 1$.

This is exactly P($k+1$). So, P(k) → P($k+1$).

So, the claim is true for all natural numbers by induction.

We know (by IH)...

$3 \mid 2^{2k-1} - 1$

...which means...

$2^{2k-1} - 1 = 3j$

We're trying to get...

$3 \mid 2^{2(k+1)-1} - 1$

...which is true if...

$2^{2(k+1)-1} - 1 = 3k$