BILL, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

Primitive Recursive Functions and Ramsey Theory

Exposition by William Gasarch-U of MD

Bounds on a-ary Ramsey Numbers

Def $R_a(k)$ is the least n such that, for all $\mathrm{COL}\colon \binom{[n]}{a} \rightarrow [2]$ there exists a homog set of size k.

Bounds on *a*-ary Ramsey Numbers

Def $R_a(k)$ is the least n such that, for all $\mathrm{COL}\colon \binom{[n]}{a} \rightarrow [2]$ there exists a homog set of size k.

Recall that we showed
$R_2(k) \le 2^{2k-1}$.
$R_3(k) \le \mathrm{TOW}(2k)$.

Bounds on a-ary Ramsey Numbers

Def $R_a(k)$ is the least n such that, for all $\mathrm{COL}\colon \binom{[n]}{a} \rightarrow [2]$ there exists a homog set of size k.

Recall that we showed
$R_2(k) \le 2^{2k-1}$.
$R_3(k) \le \mathrm{TOW}(2k)$.

What would the bound be on $R_4(k)$?
We do not have a good way to write it down (not quite true–see Knuth's Arrow Notation).

Bounds on a-ary Ramsey Numbers

Def $R_a(k)$ is the least n such that, for all $\mathrm{COL}\colon \binom{[n]}{a} \rightarrow [2]$ there exists a homog set of size k.

Recall that we showed
$R_2(k) \le 2^{2k-1}$.
$R_3(k) \le \mathrm{TOW}(2k)$.

What would the bound be on $R_4(k)$?
We do not have a good way to write it down (not quite true–see Knuth's Arrow Notation).

Consider the function
(a, k) maps to $R_a(k)$.
What are the bounds on that?

Bounds on a-ary Ramsey Numbers

Def $R_a(k)$ is the least n such that, for all $\mathrm{COL}\colon \binom{[n]}{a} \rightarrow [2]$ there exists a homog set of size k.

Recall that we showed
$R_2(k) \le 2^{2k-1}$.
$R_3(k) \le \mathrm{TOW}(2k)$.

What would the bound be on $R_4(k)$?
We do not have a good way to write it down (not quite true–see Knuth's Arrow Notation).

Consider the function
(a, k) maps to $R_a(k)$.
What are the bounds on that?

We need a way to express very fast growing functions.

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

1. $f(x_1, \ldots, x_n) = 0$;

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

1. $f(x_1, \ldots, x_n) = 0$;
2. $f(x_1, \ldots, x_n) = x_i$;

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

1. $f(x_1, \ldots, x_n) = 0$;
2. $f(x_1, \ldots, x_n) = x_i$;
3. $f(x_1, \ldots, x_n) = x_i + 1$;

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

1. $f(x_1, \ldots, x_n) = 0$;
2. $f(x_1, \ldots, x_n) = x_i$;
3. $f(x_1, \ldots, x_n) = x_i + 1$;
4. $g_1(x_1, \ldots, x_k)$, ..., $g_n(x_1, \ldots, x_k)$,$h(x_1, \ldots, x_n)$ PR $\implies$

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_n(x_1, \ldots, x_k)) \text{ is PR}$$

Definition of Primitive Recursive (PR)

Def $f(x_1, \ldots, x_n)$ is **PR** if either:

1. $f(x_1, \ldots, x_n) = 0$;
2. $f(x_1, \ldots, x_n) = x_i$;
3. $f(x_1, \ldots, x_n) = x_i + 1$;
4. $g_1(x_1, \ldots, x_k)$, …, $g_n(x_1, \ldots, x_k)$,$h(x_1, \ldots, x_n)$ PR $\implies$

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_n(x_1, \ldots, x_k)) \text{ is PR}$$

5. $h(x_1, \ldots, x_{n+1})$ and $g(x_1, \ldots, x_{n-1})$ PR $\implies$

$$\begin{aligned} f(x_1, \ldots, x_{n-1}, 0) &= g(x_1, \ldots, x_{n-1}) \\ f(x_1, \ldots, x_{n-1}, m+1) &= h(x_1, \ldots, x_{n-1}, m, f(x_1, \ldots, x_{n-1}, m)) \end{aligned}$$

is PR.

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

$f_1(x, y) = x + y$

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

$f_1(x, y) = x + y$
$f_1(x, 0) = x$
$f_1(x, y + 1) = f_1(x, y) + 1$.
Used Rec Rule Once. Addition.

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

$f_1(x, y) = x + y$
$f_1(x, 0) = x$
$f_1(x, y + 1) = f_1(x, y) + 1$.
Used Rec Rule Once. Addition.

$f_2(x, y) = xy$:

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

$f_1(x, y) = x + y$
$f_1(x, 0) = x$
$f_1(x, y + 1) = f_1(x, y) + 1$.
Used Rec Rule Once. Addition.

$f_2(x, y) = xy$:
$f_2(x, 1) = x$ (Didn't start at 0. A detail.)
$f_2(x, y + 1) = f_2(x, y) + x$.
Used Rec Rule Twice. Once to get $x + y$ PR, and once here.
Multiplication

Examples of PR Functions

$f_0(x, y) = y + 1$. Successor.

$f_1(x, y) = x + y$
$f_1(x, 0) = x$
$f_1(x, y + 1) = f_1(x, y) + 1$.
Used Rec Rule Once. Addition.

$f_2(x, y) = xy$:
$f_2(x, 1) = x$ (Didn't start at 0. A detail.)
$f_2(x, y + 1) = f_2(x, y) + x$.
Used Rec Rule Twice. Once to get $x + y$ PR, and once here.
Multiplication

The PR functions can be put in a hierarchy depending on how many times the recursion rule is used to build up to the function.

More PR Functions

More PR Functions

$f_3(x, y) = x^y$:

More PR Functions

$f_3(x, y) = x^y$:
$f_3(x, 0) = 1$
$f_3(x, y + 1) = f_3(x, y)x$.
Used Rec Rule three times. Exp.

More PR Functions

$f_3(x, y) = x^y$:
$f_3(x, 0) = 1$
$f_3(x, y + 1) = f_3(x, y)x$.
Used Rec Rule three times. Exp.

$f_4(x, y) = \mathrm{TOW}(x, y)$.

More PR Functions

$f_3(x, y) = x^y$:
$f_3(x, 0) = 1$
$f_3(x, y + 1) = f_3(x, y)x$.
Used Rec Rule three times. Exp.

$f_4(x, y) = \mathrm{TOW}(x, y)$.
$f_4(x, 0) = 1$
$f_4(x, y + 1) = f_4(x, y)^x$.
Used Rec Rule four times. TOWER.

More PR Functions

$f_3(x, y) = x^y$:
$f_3(x, 0) = 1$
$f_3(x, y + 1) = f_3(x, y)x$.
Used Rec Rule three times. Exp.

$f_4(x, y) = \mathrm{TOW}(x, y)$.
$f_4(x, 0) = 1$
$f_4(x, y + 1) = f_4(x, y)^x$.
Used Rec Rule four times. TOWER.

$f_5(x, y) =$ WHAT SHOULD WE CALL THIS?

More PR Functions

$f_3(x, y) = x^y$:
$f_3(x, 0) = 1$
$f_3(x, y + 1) = f_3(x, y)x$.
Used Rec Rule three times. Exp.

$f_4(x, y) = \mathrm{TOW}(x, y)$.
$f_4(x, 0) = 1$
$f_4(x, y + 1) = f_4(x, y)^x$.
Used Rec Rule four times. TOWER.

$f_5(x, y) =$ WHAT SHOULD WE CALL THIS?
$f_5(x, 0) = 1$
$f_5(x, y + 1) = \mathrm{TOW}(f_5(x, y), x)$.
Used Rec Rule five times.
What should we call this? Discuss

More PR Functions

$f_3(x,y) = x^y$:
$f_3(x,0) = 1$
$f_3(x,y+1) = f_3(x,y)x$.
Used Rec Rule three times. Exp.

$f_4(x,y) = \mathrm{TOW}(x,y)$.
$f_4(x,0) = 1$
$f_4(x,y+1) = f_4(x,y)^x$.
Used Rec Rule four times. TOWER.

$f_5(x,y) =$ WHAT SHOULD WE CALL THIS?
$f_5(x,0) = 1$
$f_5(x,y+1) = \mathrm{TOW}(f_5(x,y),x)$.
Used Rec Rule five times.
What should we call this? Discuss
Its been called WOWER (in Graham-Rothchild-Spencer Ramsey Theory Book).

The Functions That Have No Name

$f_a(x, y)$ is defined as

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y+1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition
f_2 is Multiplication

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition
f_2 is Multiplication
f_3 is Exp

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition
f_2 is Multiplication
f_3 is Exp
f_4 is Tower (This name has become standard.)

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition
f_2 is Multiplication
f_3 is Exp
f_4 is Tower (This name has become standard.)
f_5 is Wower (This name is not standard.)

The Functions That Have No Name

$f_a(x, y)$ is defined as

$f_a(x, 0) = 1$
$f_a(x, y + 1) = f_{a-1}(f_a(x, y), x, y)$
f_0 is Successor
f_1 is Addition
f_2 is Multiplication
f_3 is Exp
f_4 is Tower (This name has become standard.)
f_5 is Wower (This name is not standard.)
f_6 and beyond have no name.

Levels

Def PR_a is the set of PR functions that can be defined with $\leq a$ uses of the Recursion rule.

Levels

Def PR_a is the set of PR functions that can be defined with $\leq a$ uses of the Recursion rule.

Note One can show that any finite number of exponentials is in PR_3.

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \leq 2^{2k} = f_3(O(k))$. Level 3.

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \le 2^{2k} = f_3(O(k))$. Level 3.

$R_3(k) \le \mathrm{TOW}(2k) = f_4(O(k))$. Level 4.

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \le 2^{2k} = f_3(O(k))$. Level 3.

$R_3(k) \le \mathrm{TOW}(2k) = f_4(O(k))$. Level 4.

$R_a(k) \le f_{a+1}(O(k))$. Level $a+1$.

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \le 2^{2k} = f_3(O(k))$. Level 3.

$R_3(k) \le \mathrm{TOW}(2k) = f_4(O(k))$. Level 4.

$R_a(k) \le f_{a+1}(O(k))$. Level $a+1$.

I can now state my questions and add some more.

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \le 2^{2k} = f_3(O(k))$. Level 3.

$R_3(k) \le \mathrm{TOW}(2k) = f_4(O(k))$. Level 4.

$R_a(k) \le f_{a+1}(O(k))$. Level $a+1$.

I can now state my questions and add some more.

- Is $R_3(k)$ in PR_3?

Bounding the Hypergraph Ramsey Numbers

$R_2(k) \le 2^{2^k} = f_3(O(k))$. Level 3.

$R_3(k) \le \mathrm{TOW}(2k) = f_4(O(k))$. Level 4.

$R_a(k) \le f_{a+1}(O(k))$. Level $a+1$.

I can now state my questions and add some more.

- Is $R_3(k)$ in PR_3?
- Is the function $f(a,k) = R_a(k)$ PR?

More is PR than you Think

The following are PR:

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.
3. $f(x, y) =$ the remainder when you divide x by y.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.
3. $f(x, y) =$ the remainder when you divide x by y.
4. $f(x, y) = x \pmod{y}$.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.
3. $f(x, y) =$ the remainder when you divide x by y.
4. $f(x, y) = x \pmod{y}$.
5. $f(x, y) = \mathrm{GCD}(x, y)$.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.
3. $f(x, y) =$ the remainder when you divide x by y.
4. $f(x, y) = x \pmod{y}$.
5. $f(x, y) = \mathrm{GCD}(x, y)$.
6. $f(x) = 1$ if x is prime, 0 if not.

More is PR than you Think

The following are PR:

1. $f(x, y) = x - y$ if $x \geq y$, 0 otherwise.
2. $f(x, y) =$ the quotient when you divide x by y.
3. $f(x, y) =$ the remainder when you divide x by y.
4. $f(x, y) = x \pmod{y}$.
5. $f(x, y) = \mathrm{GCD}(x, y)$.
6. $f(x) = 1$ if x is prime, 0 if not.
7. $f(x) = 1$ if x is the sum of 2 primes, 0 otherwise.

Most Functions are PR

Virtually any computable function from N^k to N that you encounter in mathematics is primitive recursive.

Most Functions are PR

Virtually any computable function from N^k to N that you encounter in mathematics is primitive recursive.

Are there any computable functions that are not primitive recursive?
Discuss.

Most Functions are PR

Virtually any computable function from N^k to N that you encounter in mathematics is primitive recursive.

Are there any computable functions that are not primitive recursive?
Discuss.

Yes. We will see a contrived one on the next slide.

A Contrived Not PR Function

The PR functions are formed by building up rules. One can encode the derivation of a PR function as a number. One can then assign to every number a PR function easily.

A Contrived Not PR Function

The PR functions are formed by building up rules. One can encode the derivation of a PR function as a number. One can then assign to every number a PR function easily.

Let $f_1, f_2, \ldots$ be all of the PR functions.

A Contrived Not PR Function

The PR functions are formed by building up rules. One can encode the derivation of a PR function as a number. One can then assign to every number a PR function easily.

Let $f_1, f_2, \ldots$ be all of the PR functions.

$$F(x) = f_x(x) + 1$$

is computable but not a PR function.

A "Natural" non PR Function

Def **Ackermann's function** is the function defined by

$$
\begin{aligned}
A(0, y) &= y + 1 \\
A(x + 1, 0) &= A(x, 1) \\
A(x + 1, y + 1) &= A(x, A(x + 1, y))
\end{aligned}
$$

A "Natural" non PR Function

Def **Ackermann's function** is the function defined by

$$\begin{aligned} A(0, y) &= y + 1 \\ A(x+1, 0) &= A(x, 1) \\ A(x+1, y+1) &= A(x, A(x+1, y)) \end{aligned}$$

1. A is obviously computable.

A "Natural" non PR Function

Def **Ackermann's function** is the function defined by

$$\begin{aligned} A(0,y) &= y+1 \\ A(x+1,0) &= A(x,1) \\ A(x+1,y+1) &= A(x,A(x+1,y)) \end{aligned}$$

1. A is obviously computable.
2. If f is prim rec then it is defined by 8 recursions. Or 18. Or any constant number. But $A(x,y)$ uses y recursions, not a constant.

A "Natural" non PR Function

Def **Ackermann's function** is the function defined by

$$\begin{aligned} A(0,y) &= y+1 \\ A(x+1,0) &= A(x,1) \\ A(x+1,y+1) &= A(x,A(x+1,y)) \end{aligned}$$

1. A is obviously computable.
2. If f is prim rec then it is defined by 8 recursions. Or 18. Or any constant number. But $A(x,y)$ uses y recursions, not a constant.
3. A grows faster than any PR function.

A "Natural" non PR Function

Def **Ackermann's function** is the function defined by

$$\begin{aligned} A(0,y) &= y+1 \\ A(x+1,0) &= A(x,1) \\ A(x+1,y+1) &= A(x,A(x+1,y)) \end{aligned}$$

1. A is obviously computable.
2. If f is prim rec then it is defined by 8 recursions. Or 18. Or any constant number. But $A(x,y)$ uses y recursions, not a constant.
3. A grows faster than any PR function.
4. Since A is defined using a recursion which involves applying the function to itself there is no obvious way to take the definition and make it PR. Not a proof, an intuition.

Ackermann's Function is Natural: Security

https://www.ackermansecurity.com/

Ackermann's Function is Natural: Security

`https://www.ackermansecurity.com/`

They are called Ackerman Security since they claim that a thief would have to take time Ackerman(n) to break in.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

(2) If A, B are sets then make $A \cup B$ a set.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

(2) If A, B are sets then make $A \cup B$ a set.

(3) Given x find which, if any, set it is in.

Ackermann's Function is Natural: DS

DS is Data Structure.

A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

(2) If A, B are sets then make $A \cup B$ a set.

(3) Given x find which, if any, set it is in.

- There is a DS for this problem that can do n operations in $nA^{-1}(n, n)$ steps.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

(2) If A, B are sets then make $A \cup B$ a set.

(3) Given x find which, if any, set it is in.

- There is a DS for this problem that can do n operations in $nA^{-1}(n, n)$ steps.
- One can show that there is no better DS.

Ackermann's Function is Natural: DS

DS is Data Structure.
A **Union-Find DS** for sets supports the following operations.

(1) If a is a number then make $\{a\}$ a set.

(2) If A, B are sets then make $A \cup B$ a set.

(3) Given x find which, if any, set it is in.

- There is a DS for this problem that can do n operations in $nA^{-1}(n, n)$ steps.
- One can show that there is no better DS.

So $nA^{-1}(n, n)$ is the exact upper and lower bound!

More Natural Examples of Non-Prim Rec Fns

More natural examples of non-prim rec functions:

More Natural Examples of Non-Prim Rec Fns

More natural examples of non-prim rec functions:

1. Goodstein Sequences (next slide packet).

More Natural Examples of Non-Prim Rec Fns

More natural examples of non-prim rec functions:

1. Goodstein Sequences (next slide packet).
2. Finite Version of Kruskal's Tree Theorem.

Ackermann's Function & Goodstein Sequences

Writing a number as a sum of powers of 2.

$$1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

Ackermann's Function & Goodstein Sequences

Writing a number as a sum of powers of 2.

$$1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

But we can also write the exponents as sums of powers of 2

$$1000 = 2^{2^3+2^0} + 2^{2^3} + 2^{2^2+2^1+2^0} + 2^{2^2+2^1} + 2^{2^2+2^0} + 2^{2^1+2^0}$$

Ackermann's Function & Goodstein Sequences

Writing a number as a sum of powers of 2.

$$1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

But we can also write the exponents as sums of powers of 2

$$1000 = 2^{2^3+2^0} + 2^{2^3} + 2^{2^2+2^1+2^0} + 2^{2^2+2^1} + 2^{2^2+2^0} + 2^{2^1+2^0}$$

We can even write the exponents that are not already powers of 2 as sums of powers of 2.

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

Ackermann's Function & Goodstein Sequences

Writing a number as a sum of powers of 2.

$$1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

But we can also write the exponents as sums of powers of 2

$$1000 = 2^{2^3+2^0} + 2^{2^3} + 2^{2^2+2^1+2^0} + 2^{2^2+2^1} + 2^{2^2+2^0} + 2^{2^1+2^0}$$

We can even write the exponents that are not already powers of 2 as sums of powers of 2.

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

This is called **Hereditary Base *n* Notation**

Ackermann's Function and Goodstein Seq

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

Replace all of the 2's with 3's:

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0}$$

Ackermann's Function and Goodstein Seq

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

Replace all of the 2's with 3's:

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0}$$

This number just went WAY up. Now subtract 1.

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0} - 1$$

Ackermann's Function and Goodstein Seq

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

Replace all of the 2's with 3's:

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0}$$

This number just went WAY up. Now subtract 1.

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0} - 1$$

Repeat the process:
Replace 3 by 4, and subtract 1, Replace 4 by 5, and subtract 1, $\cdots$.

Ackermann's Function and Goodstein Seq

$$1000 = 2^{2^{2^{2^0}+2^0}+2^0} + 2^{2^{2^1+2^0}} + 2^{2^2+2^{2^0}+2^0} + 2^{2^2+2^{2^0}} + 2^{2^2+2^0} + 2^{2^{2^0}+2^0}$$

Replace all of the 2's with 3's:

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0}$$

This number just went WAY up. Now subtract 1.

$$1000 = 3^{3^{3^{3^0}+3^0}+3^0} + 3^{3^{3^1+3^0}} + 3^{3^3+3^{3^0}+3^0} + 3^{3^3+3^{3^0}} + 3^{3^3+3^0} + 3^{3^{3^0}+3^0} - 1$$

Repeat the process:

Replace 3 by 4, and subtract 1, Replace 4 by 5, and subtract 1, $\cdots$.

Vote Does the sequence:

- Goto infinity (and if so how fast- perhaps Ack-like?)
- Eventually stabilizes (e.g., goes to 18 and then stops there)
- Cycles- goes UP then DOWN then UP then DOWN ....

The Sequence. . .

The Sequence. . .

goes to 0.

The Sequence...

goes to 0.

The number of steps for n to goto 0 is **much bigger** than $A(n,n)$.

Vote

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN
   YES

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN
   YES We will show $R_3(k) \le 2^{2^{O(k)}}$.

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN
   YES We will show $R_3(k) \leq 2^{2^{O(k)}}$.
2. $R_a(k)$ is PR.
   YES, NO, UNKNOWN

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN
   YES We will show $R_3(k) \le 2^{2^{O(k)}}$.
2. $R_a(k)$ is PR.
   YES, NO, UNKNOWN
   YES

Vote

1. $R_3(k)$ is in PR_3 (finite stack-of-2's rather than TOW)
   YES, NO, UNKNOWN
   YES We will show $R_3(k) \le 2^{2^{O(k)}}$.
2. $R_a(k)$ is PR.
   YES, NO, UNKNOWN
   YES We will "show" $R_a(k)$ is $\le$ stack-of-$(a-1)$ 2's.