AKS: um algoritmo para identificar números primos

Manuela da Silva Souza & Tamires Silva dos Santos

IM-UFBA

08 de novembro de 2016

Índice

Introdução

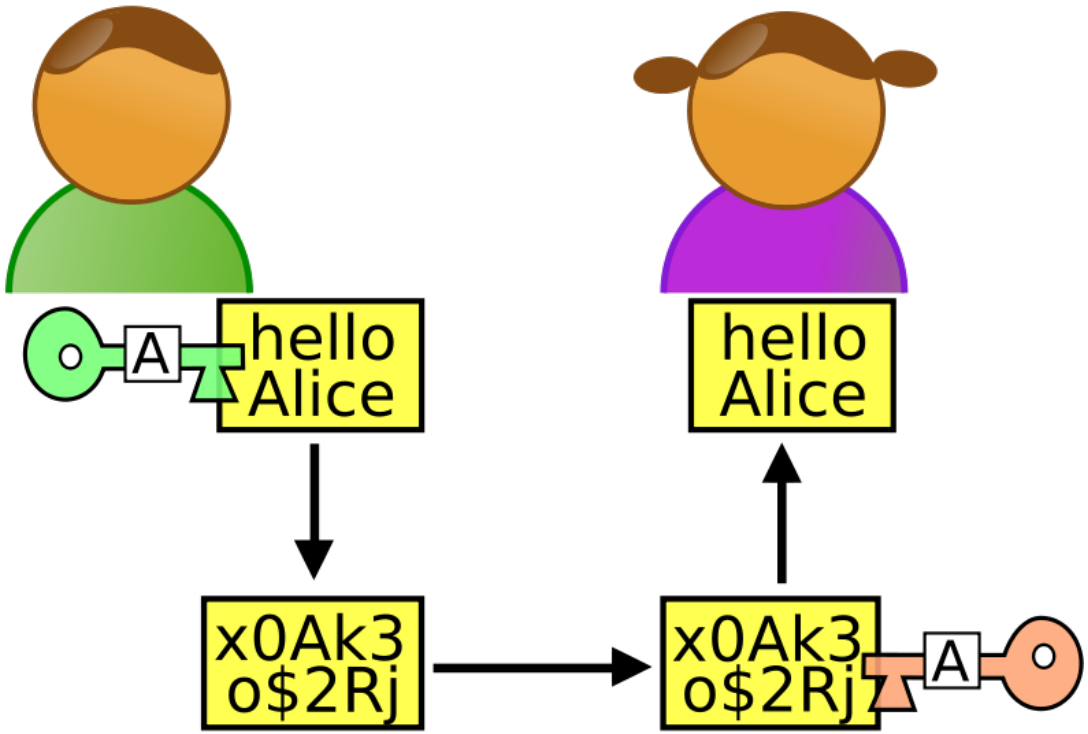
Números primos

Comprovar a primalidade de um número natural significa comprovar que este número só admite como divisores positivos ele próprio e o número um.

Motivação

A importância de saber detectar se um número é primo ou não se deve principalmente aos sistemas de criptografia, onde existe uma necessidade de gerar números primos com ordem de grandeza acima de 100 dígitos, que é vital para a segurança dos criptosistemas.

Criptografia



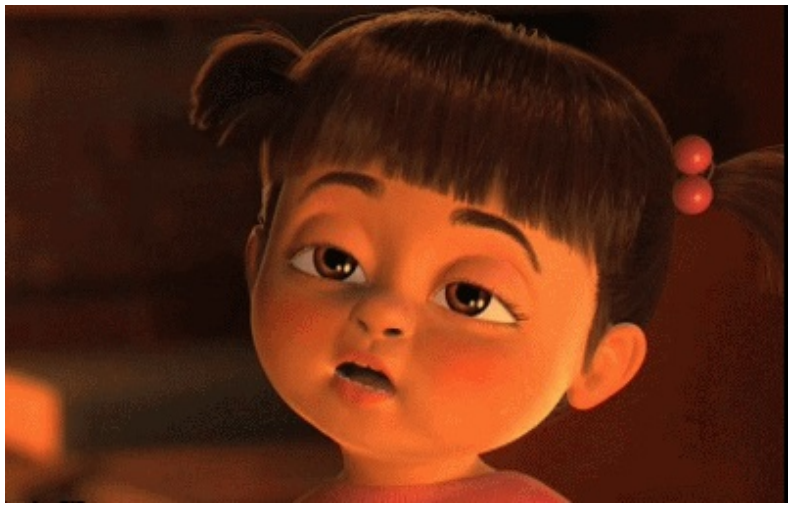
Algoritmo

O que é um algoritmo?

Um algoritmo é uma receita que mostra os procedimentos nescessários passo a passo para a resolução de uma tarefa. Tecnicamente falando, é um procedimento computacional definido que recebe um ou mais valores (entrada) e produz um ou mais valores (saída).

Exemplos de algoritmos

- Podemos dizer que acordar é um algoritmo.

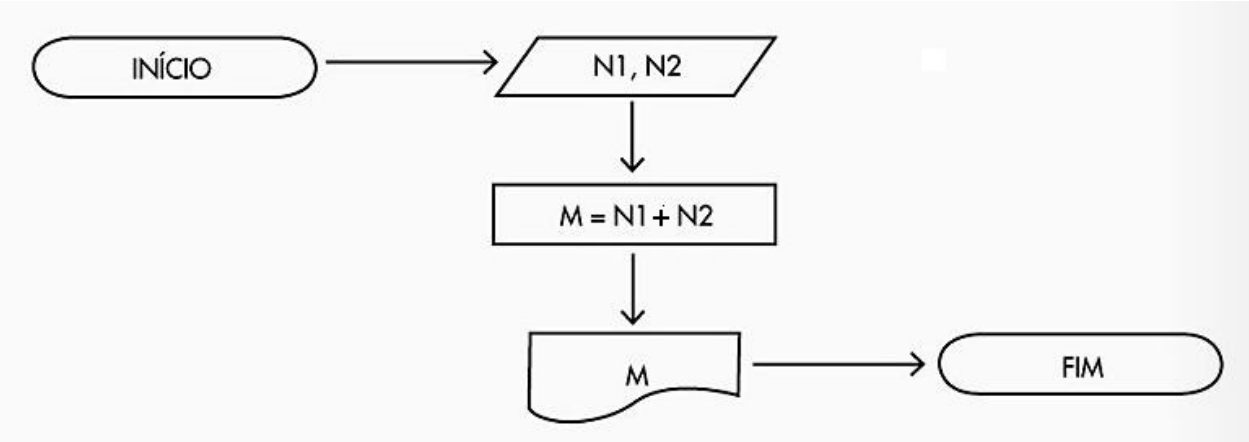


Exemplos de algoritmos

- Receita de bolo.

Exemplos de algoritmos

- Somar dois inteiros.



Bits (Dígito binário)

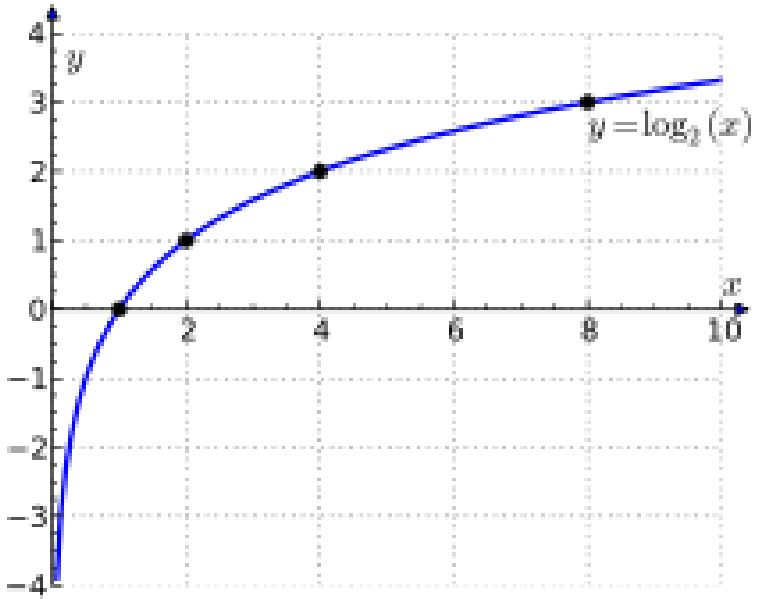
Tabela: Sistema binário

n	$(n)_2$	Quantidade de bits
0	0	1 bits
1	1	1 bits
2	10	2 bits
3	11	2 bits
4	100	3 bits
5	101	3 bits
6	110	3 bits
7	111	3 bits
8	1000	4 bits

$$n \text{ tem } k \text{ bits} \iff 2^{k-1} \leq n < 2^k \iff k - 1 \leq \log_2 n < k$$

$$k \approx \log_2 n$$

Quantidade de bits em função do número



Custo de um algoritmo

Custo de um algoritmo

O custo de um algoritmo é baseado no **tempo** que ele demora pra ser executado e a memória que ele gasta no computador.

Uma boa medida do custo para os algoritmos que estamos considerando é a quantidade de operações aritméticas elementares que são realizadas ao executá-lo, pensando em termos de bits.

Exemplo

Sejam $A, B \in \mathbb{N}$

Qual o custo da operação $A + B$?

$$A = (a_{k-1}, ..., a_1, a_0)_2, \quad B = (b_{k-1}, ..., b_1, b_0)_2$$

Ordem de crescimnento

$f(k) =$ função custo

$$f(k) = a_n k^n + \cdots + a_0 \Rightarrow f(k) \text{ é } \mathcal{O}(k^n)$$

Exemplos

1. O custo da soma $= 2k - 1$ é $\mathcal{O}(k)$.
2. O custo do produto $= 4k^2 - k$ é $\mathcal{O}(k^2)$.

Ordem de crescimnento

Limite assintótico superior ($\mathcal{O}$)

$f(k)$ é $\mathcal{O}(g(k))$ se existe constante c tal que

$$f(k) \leq c.g(k)$$

para k suficientemente grande.

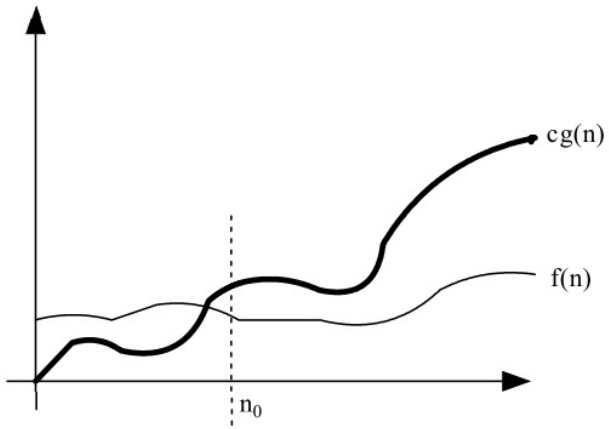


Tabela dos custos

Tabela: Custo das operações principais

Operação	Custo
Soma	$\mathcal{O}(k)$
Subtração	$\mathcal{O}(k)$
Multiplicação	$\mathcal{O}(k^2)$
Divisão	$\mathcal{O}(k^2)$

Testes de primalidade

Teste de primalidade

Chamamos de teste de primalidade um algoritmo onde a entrada é um número natural e a saída é a classificação, se esse número é primo ou composto.

Classificação dos testes de primalidade

Os testes de primalidade podem ser classificados quanto ao **custo** (polinomial ou não polinomial) e quanto a **exatidão** (determinístico ou não determinístico).

Teste polinomial e não polinomial

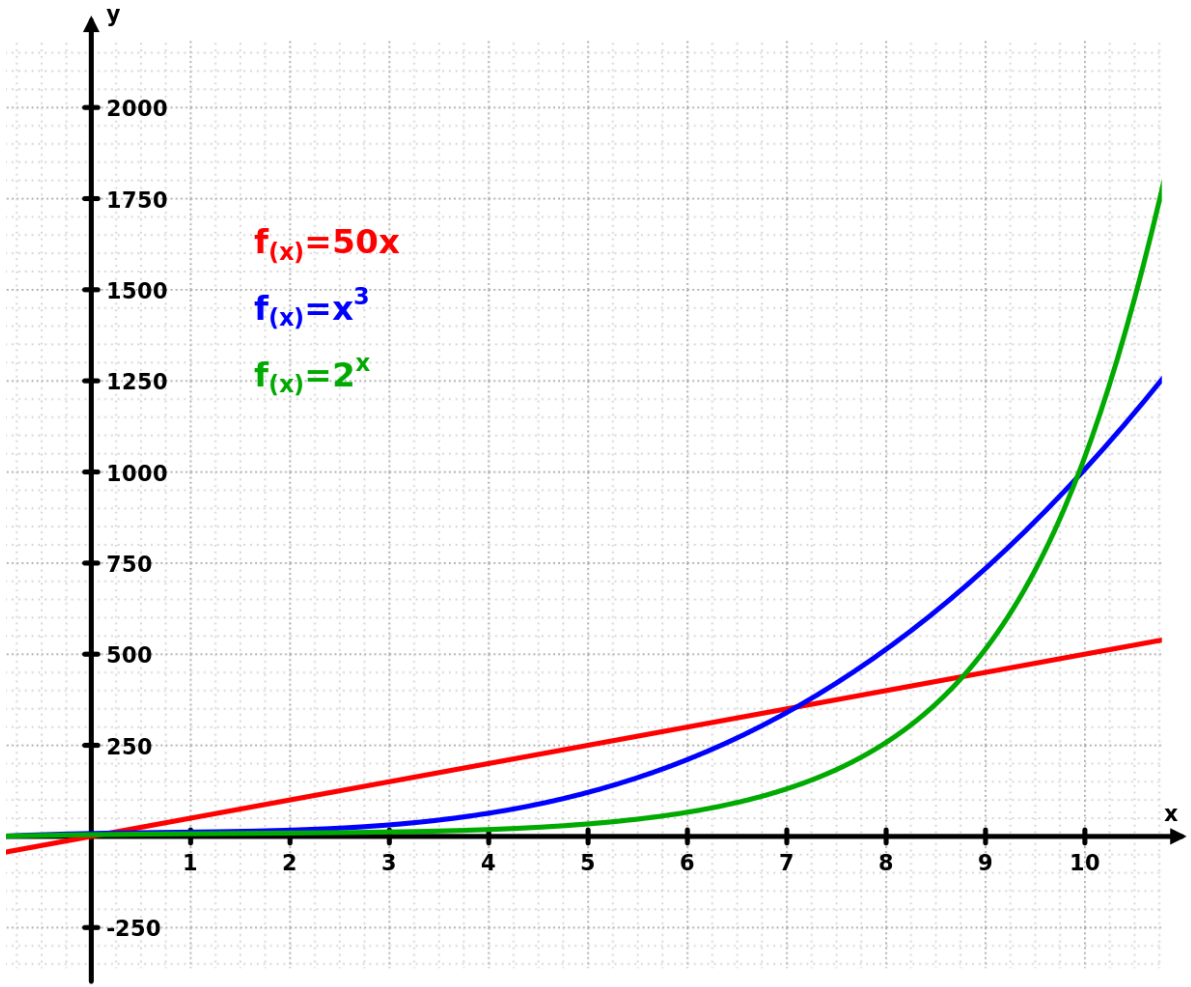
Teste de primalidade polinomial

É um teste de primalidade onde o custo pode ser escrito como um polinômio em função de k de grau r, para algum inteiro $r > 0$.

Teste de primalidade não polinomial

É um teste de primalidade onde o custo **não pode** ser escrito como um polinômio em função de k de grau r, para algum inteiro $r > 0$.

Teste polinomial e não polinomial



Testes determinístico e não determinístico

Teste de primalidade deterministico

É um teste de primalidade onde temos a garantia de que o resultado está correto, seja ele primo ou composto.

Teste de primalidade não determinístico

É um teste que garante apenas que o número é primo com uma certa probabilidade.

Algoritmo de Fatoração

Algoritmo de Fatoração

Entrada: inteiro positivo $n \geq 2$.
Saıda: inteiro positivo f que é o menor fator primo de n ou uma mensagem indicando que n é primo.
Primeira etapa: comece fazendo $F = 2$.
Segunda etapa: se $F > \sqrt{n}$, escreva n é primo e pare.
Terceira etapa: se n/F é inteiro escreva "F é fator de n"e pare; senão, incremente F de uma unidade e volte a segunda etapa.

- O custo é $\mathcal{O}(2^{k/2}(k^2))$.
- É determinístico.

Crivo de Eratóstenes

Crivo de Eratóstenes

A primeira tentativa de teste de primalidade foi o Crivo de Eratóstenes, um método semelhante a uma peneira para determinar quais eram os números primos anteriores a um outro número dado.

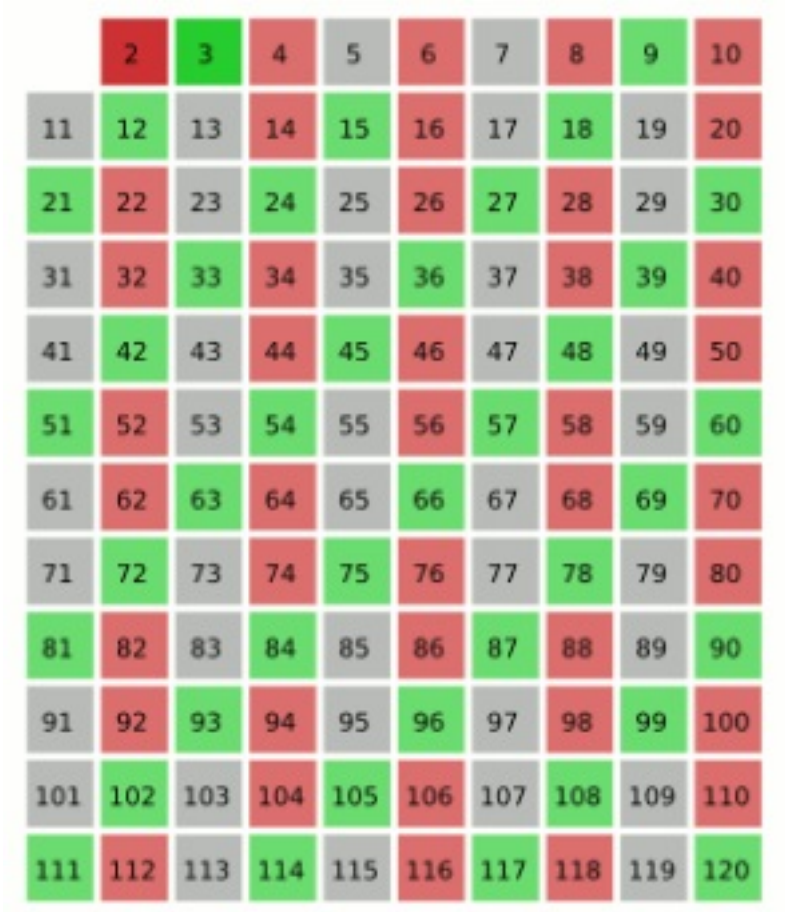
Crivo de Eratóstenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Crivo de Eratóstenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Crivo de Eratóstenes



Crivo de Eratóstenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Crivo de Eratóstenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Primos

2	3	5	7
11	13	17	19
23	29	31	37
41	43	47	53
59	61	67	71
73	79	83	89
97	101	103	107
109	113		

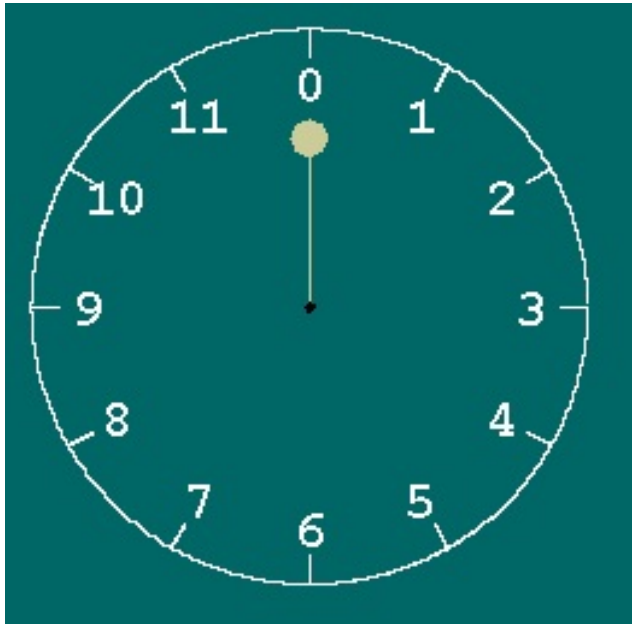
Crivo de Eratóstenes

Crivo de Eratóstenes

Entrada: inteiro positivo $n \geq 2$.
Saída: todos os primos entre 1 e o número digitado.
Primeira etapa: $F = \sqrt{n}$.
Segunda etapa: criar um vetor com os números de 2 até n.
Terceira etapa: para $i = 2$ até F, caso o número i não esteja riscado insira-o na lista dos primos, imprima-o e risque todos os seus múltiplos na lista. Faça $i = i + 1$.

- O custo é $\mathcal{O}(2^{3k/2}(k))$
- É determinístico.

Aritmética do relógio



Onde estará o ponteiro do relógio quando se passarem:

1. 32 horas
2. 89 horas
3. 56 horas

Aritmética do relógio

$$\overline{0} : 0, 12, 24, ...$$

$$\overline{1} : 1, 13, 25, ...$$

$$\overline{2} : 2, 14, 26, ...$$

$$\overline{3} : 3, 15, 27, ...$$

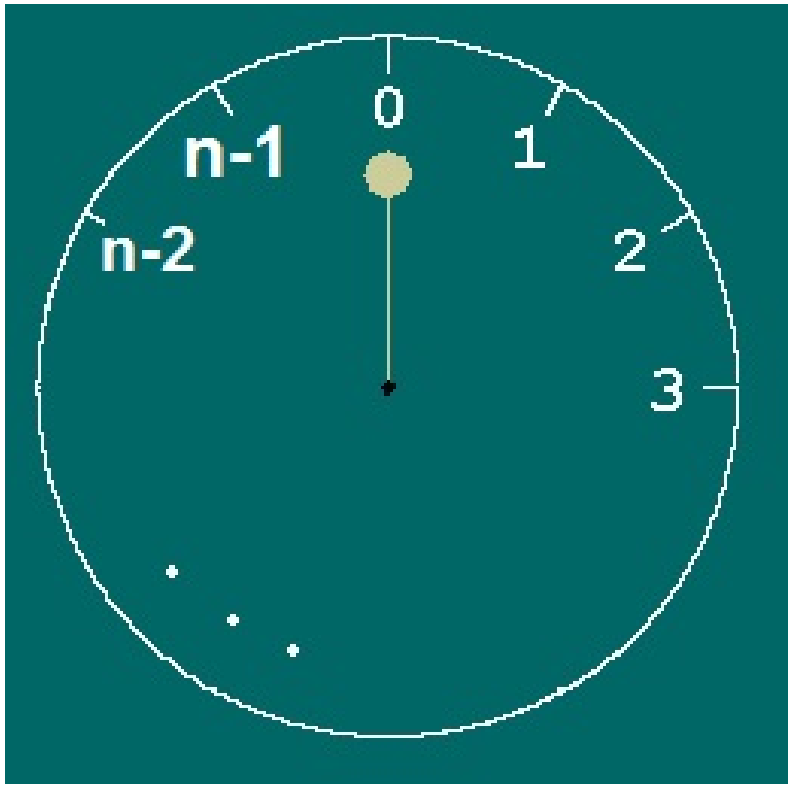
$$\overline{4} : 4, 16, 28, ...$$

$$\dots$$

$$\overline{11} : 11, 23, 35, ...$$

$$\mathbb{Z}_{12} = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \cdots, \overline{11}\}$$

Aritmética do relógio generalizado



$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \cdots, \overline{n-1}\}$$

Teste de primalidade determinístico

Teorema

Sejam n e b inteiros positivos e primos entre si. Então,

$$(x + \overline{b})^n = (x^n + \overline{b})$$

em $\mathbb{Z}_n[x]$, se, e somente se, n é primo.

- $(x + \overline{1})^2 = x^2 + 2.x.\overline{1} + \overline{1}^2 = x^2 + \overline{2}x + \overline{1}^2 = x^2 + \overline{1}.$
- $(x + \overline{\overline{1}})^4 = x^4 + \overline{\overline{2}}x^2 + \overline{\overline{1}} \neq x^4 + \overline{\overline{1}}.$

Ideia da demonstração

- Desenvolvendo o binômio, temos que:

$$(x+\overline{b})^n = x^n + \overline{b}^n + \sum_{k=1}^{n-1} \binom{n}{k} x^{n-k} b^k$$

em que, $\binom{n}{k} = \frac{n!}{(n-k)!k!}$.

- Isso pode ser provado usando o fato que

$$\binom{n}{k} \equiv 0 \pmod{n}$$

para todo $0 < k < n$ se, e somente se, n é primo.

Triângulo de Pascal

0	$\binom{0}{0}$	1
1	$\binom{1}{0}\ \binom{1}{1}$	1 1
2	$\binom{2}{0}\ \binom{2}{1}\ \binom{2}{2}$	1 2 1
3	$\binom{3}{0}\ \binom{3}{1}\ \binom{3}{2}\ \binom{3}{3}$	1 3 3 1
4	$\binom{4}{0}\ \binom{4}{1}\ \binom{4}{2}\ \binom{4}{3}\ \binom{4}{4}$	1 4 6 4 1
5	$\binom{5}{0}\ \binom{5}{1}\ \binom{5}{2}\ \binom{5}{3}\ \binom{5}{4}\ \binom{5}{5}$	1 5 10 10 5 1
6	$\binom{6}{0}\ \binom{6}{1}\ \binom{6}{2}\ \binom{6}{3}\ \binom{6}{4}\ \binom{6}{5}\ \binom{6}{6}$	1 6 15 20 15 6 1

Ideia da demonstração

- Usando o resultado anterior, temos que:

$$(x + \overline{b})^n = x^n + \overline{b}^n.$$

- Resta mostrar que

$$\overline{b}^n = \overline{b} \pmod{n}.$$

Teorema de Fermat

Seja n um primo positivo e b um número inteiro. Se n não divide b, então

$$\overline{b}^{n-1} = \overline{1} \pmod{n}.$$

O Algoritmo determinístico

Teste de primalidade determinístico

Entrada: inteiro positivo n.
Saıda: mensagem indicando se n é composto ou n é primo.
Primeira etapa: calcule todos os termos da expansão $(x + \overline{b})^n$ em $\mathbb{Z}_n[x]$.
Segunda etapa: verifique se a expressão obtida na etapa anterior é igual a $x^n + \overline{b}$. Se for, n é primo. Se não for, n é composto.

- O custo é $\mathcal{O}(2^k)$.
- É determinístico.

Teste de primalidade não determinístico

Teorema de Fermat

Seja n um primo positivo e b um número inteiro. Se n não divide b, então

$$\overline{b}^{n-1} = \overline{1} \pmod{n}.$$

A recíproca é falsa. Por exemplo,

$$\overline{2340} = \overline{1} = \overline{2340}^{340} \pmod{341},$$

mas $341 = 11.31$.

Teste de primalidade não determinístico

Suponha n primo ímpar e $b \in \{1, ..., n-1\}$. Escreva $n - 1 = 2^k q$.

$$b^q, b^{2q}, ..., b^{2^{k-1}q}, b^{2^k q}$$

Mas $b^{2^k q} \equiv b^{n-1} \equiv 1 \pmod{n}$. Digamos que j é o menor expoente tal que $b^{2^k q} \equiv 1 \pmod{n}$.

- Se $j = 1$, significa que o primeiro termo desta sequência é congruo a 1.
- Se $j \geq 1$ podemos usar que

$$b^{2^j q} - 1 = (b^{2^{j-1}} - 1)(b^{2^{j-1}} + 1) \equiv 0 \pmod{n}$$

signifca que essa sequência tem um termo congruo a -1 em alguma de suas posições.

Teste de primalidade não determinístico

Teste de Miller-Rabin

Entrada: um inteiro ímpar n, que se quer testar, e a base b, onde $1 < b < n - 1$.
Saıda: uma das mensagens "n é composto"ou "teste inconclusivo"
Primeira etapa: divida n-1 sucessivamente por 2 até encontrar q (um número ímpar)e k tais que $n - 1 = 2^k q$.
Segunda etapa: começe fazendo i=0 e r=resíduo de b^q por n.
Terceira etapa: se $i = 0$ e $r = 1$ ou se $i \geq 0$ e $r = n - 1$ a saída é 'teste inconclusivo'.
Quarta etapa: incremente i de 1 e substitua r pelo resto da divisão de r por n.
Quinta etapa: se $i < k - 1$ volte a etapa 3, senão a saída é 'n é composto'

- O custo é $\mathcal{O}(k^3)$.
- É não determinístico.

Gary L. Millera

Michael O. Rabin

Comparação dos testes

Tabela: Testes de primalidade

Nome do Teste	Determinístico	Polinomial
Algoritmo de fatoração	Sim	Não
Crivo de Eratóstenes	Sim	Não
Teste determinístico $\mathbb{Z}_n[x]$	Sim	Não
Teste de Miller-Rabin	Não	Sim

O Algoritmo AKS

- O algoritmo AKS ganhou destaque por ser o primeiro algoritmo publicado que é simultaneamente **polinomial e determinístico**.
- A descoberta de um algoritmo polinomial de primalidade foi publicada em agosto de 2002 por Agrawal, Kayal e Saxena. Três indianos, dois dos quais haviam acabado há pouco o curso de graduacão.

O Algoritmo AKS

Figura: Manindra Agrawal, Neeraj Kayal e Nitin Saxena

O Algoritmo AKS

O teste de primalidade AKS é baseado na equivalência:

$$(x + \overline{b})^n \equiv (x^n + \overline{b}) \pmod{n}$$

se, e somente se, n é primo.

AKS faz uso da relação de equivalência

$$(x + \overline{b})^n \equiv (x^n + \overline{b}) \pmod{n, x^r - 1}$$

Esta pode ser verificada em tempo polinomial.

O Algoritmo AKS

- Contudo, enquanto todos os primos satisfazem esta equivalência, alguns números compostos também a satisfazem.
- A prova da correção consiste em mostrar que existe um r pequeno e um conveniente conjunto de inteiros A tal que se a equivalência que assegura para todo a em A, então n deve ser primo.

Tabela dos custos

Tabela: Custo das etapas do AKS

Etapa	Custo
1	$\mathcal{O}(k^7)$
2	$\mathcal{O}(k^{12})$
3	$\mathcal{O}(k^{20})$
4	$\mathcal{O}(k^{33})$

Custo AKS $= \mathcal{O}(k^7) + \mathcal{O}(k^{12}) + \mathcal{O}(k^{20}) + \mathrm{O}(\mathrm{k}^{33})$ portanto é $\mathcal{O}(k^{33})$

Bibliografia I


Agrawal, M., N. Kayal e N. Saxena
2002 Primes is in P, IIT, preprint.
disponível em http://www.cse.utk.ac.in/ news/primality.pdf


Farias, F.
Uma análise comparativa entre os testes de primalidade AKS e Miller-Rabin .
disponível em
https://www.ucb.br/sites/100/103/TCC/22007/FernandodeFarias.pd


Wikipédia
Teste de primalidade AKS .
disponível em
https://pt.wikipedia.org/wiki/Teste_de_primalidade_AKS

Bibliografia II

Coutinho S. C.
Primalidade em Tempo Polinomial: uma introdução ao algoritmo AKS.
Universidade Federal do Rio de Janeiro, 2004.

Obrigada pela atenção!