Primos, MDC e MMC

Matemática Discreta

Prof. Samy Sá

Universidade Federal do Ceará
Campus de Quixadá

16 de setembro de 2020

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Prévia

Requisitos

- Técnicas de Demonstração de Teoremas
- Propriedades de operações aritméticas
- Divisibilidade

Esta apresentação...

- Introduz números primos intuitivamente e formalmente
- Apresenta algoritmos para verificação de números primos
- Explora aplicações de números primos
- Discute os conceitos de máximo divisor comum, de mínimo múltiplo comum, e mostra várias formas de calculá-los

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que a **divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que a é um **divisor** de b se e somente se a divide b.

Então dado n inteiro, cada m que divide n satisfaz:

- $m \neq 0$,
- se $n > 0$, então $m \leq n$,
- se $n < 0$, então $m \geq n$,
- se $n \neq 0$, então $|m| \leq |n|$,
- m também divide $-n$,
- $-m$ também divide n.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que a **divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que a é um **divisor** de b se e somente se a divide b.

Exemplo

Quem são os divisores de 3? *Cada m que divide 3 satisfaz:*

- $m \neq 0$,
- *como* $3 > 0$, *então* $m \leq 3$,
- *como* $3 \neq 0$, *então* $|m| \leq 3$,
- m *também divide* -3,
- $-m$ *também divide* 3.

OU SEJA, $-3 \leq m \leq 3$ para todo inteiro m que divide 3.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que a **divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que *a* é um **divisor** de *b* se e somente se *a* divide *b*.

Exemplo

Quem são os divisores de 3? *Cada m que divide 3 satisfaz:*

- $m \neq 0$,
- *como* $3 > 0$, *então* $m \leq 3$,
- *como* $3 \neq 0$, *então* $|m| \leq 3$,
- m *também divide* -3,
- $-m$ *também divide* 3.

ENTÃO, vamos testar cada $-3 \leq m \leq 3$ ...

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que a **divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que a é um **divisor** de b se e somente se a divide b.

Exemplo

Quem são os divisores de 3?

- $-3 \mid 3$?
- $-2 \mid 3$?
- $-1 \mid 3$?
- $0 \mid 3$?
- $1 \mid 3$?
- $2 \mid 3$?
- $3 \mid 3$?

ENTÃO, vamos testar cada $-3 \leq m \leq 3$ ...

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que **a divide b** se e somente se **existe um inteiro c tal que $b = ac$**.*

LEMBRE-SE: Dizemos que *a* é um **divisor** de *b* se e somente se *a* divide *b*.

Exemplo

Quem são os divisores de 3?

- $-3 \mid 3$? ✓
- $-2 \mid 3$? x
- $-1 \mid 3$? ✓
- $0 \mid 3$? x
- $1 \mid 3$? ✓
- $2 \mid 3$? x
- $3 \mid 3$? ✓

ENTÃO, vamos testar cada $-3 \leq m \leq 3$ ...

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que a **divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que *a* é um **divisor** de *b* se e somente se *a* divide *b*.

Exemplo

Quem são os divisores de 3?

- $-3 \mid 3$? ✓
- $-2 \mid 3$? x
- $-1 \mid 3$? ✓
- $0 \mid 3$? x
- $1 \mid 3$? ✓
- $2 \mid 3$? x
- $3 \mid 3$? ✓

CONCLUSÃO: $-3, -1, 1$ e 3 (quatro divisores).

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

*Sejam a e b números inteiros com $a \neq 0$, dizemos que **a divide** b se e somente se **existe um inteiro** c **tal que** $b = ac$.*

LEMBRE-SE: Dizemos que *a* é um **divisor** de *b* se e somente se *a* divide *b*.

Exercício

Quem são os divisores de cada número abaixo?

- *3*
- *-3*
- *4*
- *-4*
- *10*
- *-10*
- *1*
- *-1*
- *0*

Alternativamente, **QUANTOS** divisores tem cada número?

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Quantos Divisores?

Definição (Divisibilidade)

Sejam a e b números inteiros com $a \neq 0$, dizemos que a ***divide*** *b se e somente se* ***existe um inteiro*** *c* ***tal que*** *$b = ac$.*

LEMBRE-SE: Dizemos que *a* é um **divisor** de *b* se e somente se *a* divide *b*.

Os tópicos desta aula giram em torno destas perguntas.

1. **QUEM** são os divisores de um inteiro?
2. **QUANTOS** são os divisores de um inteiro?

OBS.: Por simplicidade, lidaremos apenas com números positivos.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se n **tem exatamente dois divisores positivos**.*

Se um inteiro $n > 1$ não for primo, diremos que n é **composto**.

Exemplo

- 11 *é primo, pois seus divisores positivos são* 1 *e* 11 *(exatamente 2).*
- 15 *é composto, pois seus divisores positivos são* $1, 3, 5$ *e* 15 *(mais que 2).*

Números Primos

Definição (Número Primo)

Seja n um inteiro maior que 1, dizemos que n ***é primo*** *se e somente se n* ***tem exatamente dois divisores positivos****.*

Se um inteiro $n > 1$ não for primo, diremos que n é **composto**.

Note que

- Para todo $n \neq 0$, temos $1 \mid n$.
- Para todo $n \neq 0$, temos $n \mid n$.

Portanto [1/3], todo inteiro $n > 1$ **terá no mínimo dois divisores positivos**!

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se n **tem exatamente dois divisores positivos**.*

Se um inteiro $n > 1$ não for primo, diremos que n é **composto**.

Note que

- Para todo $n \neq 0$, temos $1 \mid n$.
- Para todo $n \neq 0$, temos $n \mid n$.

Portanto [2/3], um inteiro $n > 1$ **é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo)

Seja n um inteiro maior que 1, dizemos que n ***é primo*** *se e somente se n* ***tem exatamente dois divisores positivos****.*

Se um inteiro $n > 1$ não for primo, diremos que n é **composto**.

Note que

- Para todo $n \neq 0$, temos $1 \mid n$.
- Para todo $n \neq 0$, temos $n \mid n$.

Portanto [3/3], um inteiro $n > 1$ **é composto** se e somente se n **tem três ou mais divisores positivos**.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.*

Proposição ("três divisores")

Um inteiro n é composto se e somente se n tem pelo menos três divisores positivos.

Lembre-se que para todo m que divide n:

- se $n > 0$, então $m \leq n$.
- se desejamos $m > 0$, teremos $m \geq 1$.

Ou seja, é necessário que $1 \leq m \leq n$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

Seja n um inteiro maior que 1*, dizemos que* ***n é primo*** *se e somente se* ***os únicos divisores positivos de n são 1 e n****.*

Proposição (“terceiro divisor”)

Um inteiro n *é composto se e somente se* n *tem um divisor* k *tal que* $1 < k < n$*.*

Podemos fazer ainda melhor:

1. Seja n um inteiro, suponha que n é composto.
2. Pela proposiçao “terceiro divisor”, existe um inteiro k tal que $k \mid n$ e $1 < k < n$.
3. Pela definição de divisibilidade, existe um inteiro j tal que $n = k.j$.
4. Neste ponto, é necessário que que $k \leq \sqrt{n}$ ou $j \leq \sqrt{n}$.

Ou seja, ao menos um divisor de n é menor ou igual a $\sqrt{n}$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.*

Proposição ("máximo $\sqrt{n}$")

Um inteiro n é composto se e somente se n tem um divisor k tal que $1 < k \leq \sqrt{n}$.

Mais um passo!

1. Seja n um inteiro, suponha que n é composto.
2. Pela proposiçao "máximo $\sqrt{n}$", existe um inteiro k tal que $k \mid n$ e $1 < k \leq \sqrt{n}$.
3. Temos duas possibilidades: k é primo ou k é composto.

Caso 1) Suponha que k é primo.
Então n tem ao menos um divisor **primo** menor ou igual à $\sqrt{n}$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

*Seja n um inteiro maior que 1, dizemos que **n é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.*

Proposição (“máximo $\sqrt{n}$”)

Um inteiro n é composto se e somente se n tem um divisor k tal que $1 < k \leq \sqrt{n}$.

Mais um passo!

1. Seja n um inteiro, suponha que n é composto.
2. Pela proposiçao “máximo $\sqrt{n}$”, existe um inteiro k tal que $k \mid n$ e $1 < k \leq \sqrt{n}$.
3. Temos duas possibilidades: k é primo ou k é composto.

Caso 2) Suponha que k é composto. Então, pela proposição “máximo $\sqrt{n}$”, existe um inteiro j tal que $j \mid k$ e $1 < j \leq \sqrt{k}$.
Temos duas possibilidades: j é primo ou j é composto **...**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.*

Proposição ("máximo $\sqrt{n}$")

Um inteiro n é composto se e somente se n tem um divisor k tal que $1 < k \leq \sqrt{n}$.

Mais um passo!

1. Seja n um inteiro, suponha que n é composto.
2. Pela proposiçao "máximo $\sqrt{n}$", existe um inteiro k tal que $k \mid n$ e $1 < k \leq \sqrt{n}$.
3. Temos duas possibilidades: k é primo ou k é composto.

Caso 2) Suponha que k é composto.

...

Então n tem ao menos um divisor **primo** menor ou igual à $\sqrt{n}$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Números Primos

Definição (Número Primo - Alternativa)

*Seja n um inteiro maior que 1, dizemos que n **é primo** se e somente se **os únicos divisores positivos de n são 1 e n**.*

Teorema ("primo até $\sqrt{n}$")

*Um inteiro n é composto se e somente se n tem um divisor p **primo** tal que $p \leq \sqrt{n}$.*

Este teorema é fundamental para os agoritmos que estudaremos.

Como averiguar se um inteiro $n > 1$ é primo?

1. se você **já conhece** os números primos até $\sqrt{n}$, use o Algoritmo de Fatoração.
2. se você **não conhece** os números primos até $\sqrt{n}$, use o Crivo de Eratóstenes.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Idéia: Encontrar todos os primos no intervalo $1 < k \leq n$.

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até *n* a uma lista e marque-os com "?"

2. Para *k* de 2 até *n* faça

3. se *k* é "?", faça $\{$

4. marque *k* com "**primo**"

5. marque cada múltiplo de *k* com "**não-primo**"

6. $\}$

Executaremos o algoritmo para $n = 100$**.**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 1:

$k = 2$, marcado com “?”

\	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

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 1:

$k = 2$, marcado com “**primo**”

\	**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

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça $\{$
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. $\}$

Executaremos o algoritmo para $n = 100$.

Iteração 1:

$k = 2$, marcado com “**primo**”
e cada múltiplo de $k = 2$
identificado (para visualização)

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 1:

$k = 2$, marcado com "**primo**" e cada múltiplo de $k = 2$ marcado com "**não-primo**"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3.   se k é "?", faça $\{$
4.   marque k com "**primo**"
5.   marque cada múltiplo de k com "**não-primo**"
6. $\}$

Executaremos o algoritmo para $n = 100$.

Iteração 2:

$k = 3$, marcado com "?"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$**.**

Iteração 2:

$k = 3$, marcado com "**primo**"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 2:

$k = 3$, marcado com "**primo**" e cada múltiplo de $k = 3$ **identificado** (para visualização)

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 2:

$k = 3$, marcado com “**primo**” e cada múltiplo de $k = 3$ marcado com “**não-primo**”

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 3:

$k = 4$, marcado com “**não-primo**”

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 4:

$k = 5$, marcado com "?"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 4:

$k = 5$, marcado com “**primo**”

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 4:

$k = 5$, marcado com "**primo**"
e cada múltiplo de $k = 5$
identificado (para visualização)

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Iteração 4:

$k = 5$, marcado com “**primo**” e cada múltiplo de $k = 5$ marcado com “**não-primo**”

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Avançando um pouco...

Iteração 6:

$k = 7$, marcado com "**primo**"
e cada múltiplo de $k = 7$
identificado (para visualização)

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Avançando um pouco...

Iteração 6:

$k = 7$, marcado com "**primo**" e cada múltiplo de $k = 7$ marcado com "**não-primo**"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com “?”
2. Para k de 2 até n faça
3. se k é “?”, faça {
4. marque k com “**primo**”
5. marque cada múltiplo de k com “**não-primo**”
6. }

Executaremos o algoritmo para $n = 100$.

Avançando mais um pouco...

Iteração 10:

$k = 11$, marcado com “**primo**” e cada múltiplo de $k = 11$ **identificado** (para visualização)

\	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

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Avançando mais um pouco...

Iteração 10:

$k = 11$, marcado com "**primo**"
e cada múltiplo de $k = 11$
marcado com "**não-primo**"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 1)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até n faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }

Executaremos o algoritmo para $n = 100$.

Avançando até o fim...

Iteração 99:

$k = 100$, marcado com "**não-primo**"

\	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

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Na prática, a execução do algoritmo estaria completa após a Iteração 10, pois...

Teorema ("primo até $\sqrt{n}$")
Um inteiro n é composto se e somente se n tem um divisor p ***primo*** *tal que* $p \leq \sqrt{n}$.

Para verificarmos se números de 2 a 100 são primos ou compostos (nosso exemplo), nos bastará executar o Crivo de Heratóstenes com testes por primos até 10.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 2)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até $\sqrt{n}$ faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }
7. Para j de k até n faça
8. se j é "?", marque j com "**primo**".

Executaremos o algoritmo para $n = 100$.

Avançando um pouco...

Iteração 6:

$k = 7$, marcado com "**primo**" e cada múltiplo de $k = 7$ marcado com "**não-primo**"

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 2)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até $\sqrt{n}$ faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }
7. Para j de k até n faça
8. se j é "?", marque j com "**primo**".

Executaremos o algoritmo para $n = 100$.

Avançando mais um pouco...

Iteração 9:

$k = 10$, marcado com "**não-primo**".

\	**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**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Crivo de Eratóstenes

Algoritmo: Crivo de Eratóstenes (Versão 2)

1. Adicione todos os inteiros de 2 até n a uma lista e marque-os com "?"
2. Para k de 2 até $\sqrt{n}$ faça
3. se k é "?", faça {
4. marque k com "**primo**"
5. marque cada múltiplo de k com "**não-primo**"
6. }
7. Para j de k até n faça
8. se j é "?", marque j com "**primo**".

Executaremos o algoritmo para $n = 100$.

E então executamos o segundo laço...

Logo após Iteração 9:

cada número que ainda estiver marcado com "?" deverá ser marcado com "**primo**".

\	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

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Verificando se *n* é Primo

Lembre-se do que discutimos:

Como averiguar se um inteiro n é primo?

1. se você **já conhece** os números primos até $\sqrt{n}$, use o Algoritmo de Fatoração.
2. se você **não conhece** os números primos até $\sqrt{n}$, use o Crivo de Eratóstenes.

Note:

- após usarmos o Crivo de Eratóstenes, podemos nos voltar à primeira estratégia.
- como executamos o Crivo de Eratóstenes para números até 100, seremos capazes de verificar se um número é primo para números até $101^2 - 1 = 10200$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Verificando se *n* é Primo

Lembre-se do que discutimos:

Como averiguar se um inteiro n é primo?

1. se você **já conhece** os números primos até $\sqrt{n}$, use o Algoritmo de Fatoração.
2. se você **não conhece** os números primos até $\sqrt{n}$, use o Crivo de Eratóstenes.

Estratégia: Tentativa e Erro

1. Determine $\sqrt{n}$.
2. Para cada primo $p \leq \sqrt{n}$, verifique se $p \mid n$.
3. Se $p \mid n$, retorne que n não é primo.

A necessidade desta estratégia serve como base para a criptografia moderna.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Verificando se *n* é Primo

Lembre-se do que discutimos:

Como averiguar se um inteiro n é primo?

1. se você **já conhece** os números primos até $\sqrt{n}$, use o Algoritmo de Fatoração.
2. se você **não conhece** os números primos até $\sqrt{n}$, use o Crivo de Eratóstenes.

Exemplo

O número 101 é primo?

1. *Precisaremos verificar se* $p \mid 101$ *para cada inteiro* $p \leq \sqrt{101}$ *que seja primo, ou seja, para cada* $p \leq 10$ *(arredondando para baixo) que seja primo.*
2. *Há apenas quatro primos neste intervalo:* $2, 3, 5, 7$.
   - **2.1** $2 \nmid 101$, *pois* 101 ***mod*** $2 = 1$.
   - **2.2** $3 \nmid 101$, *pois* 101 ***mod*** $3 = 2$.
   - **2.3** $5 \nmid 101$, *pois* 101 ***mod*** $5 = 1$.
   - **2.4** $7 \nmid 101$, *pois* 101 ***mod*** $7 = 3$.
3. *Como nenhum primo* $p \leq \sqrt{101}$ *o divide,* 101 *é primo.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Teorema Fundamental da Aritmética

Teorema

Todo inteiro $n > 1$ pode ser escrito ***de maneira única*** *como* ***um primo*** *ou* ***o produto de dois ou mais números primos*** *escritos* ***em ordem cescente****.*

Exemplos

- $2 = 2$
- $15 = 3.5$
- $20 = 2.2.5$
- $100 = 2.2.5.5$
- $300 = 2.2.3.5.5$
- $641 = 641$
- $999 = 3.3.3.37$
- $1024 = 2.2.2.2.2.2.2.2.2.2$

Note que...

- cada primo nestas listas é um **divisor** ou **fator** do número reescrito.
- um produto de primos em ordem crescente é a **fatoração** de um inteiro.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Teorema Fundamental da Aritmética

Teorema

*Todo inteiro $n > 1$ pode ser escrito **de maneira única** como **um primo** ou **o produto de dois ou mais números primos** escritos **em ordem cescente**.*

Exemplos

- $2 = 2 = 2^1$
- $15 = 3.5 = 3^1.5^1$
- $20 = 2.2.5 = 2^2.5^1$
- $100 = 2.2.5.5 = 2^2.5^2$
- $300 = 2.2.3.5.5 = 2^2.3^1.5^2$
- $641 = 641 = 641^1$
- $999 = 3.3.3.37 = 3^3.37^1$
- $1024 = 2.2.2.2.2.2.2.2.2.2 = 2^{10}$

Note que...

- cada primo nestas listas é um **divisor** ou **fator** do número reescrito.
- um produto de primos em ordem crescente é a **fatoração** de um inteiro.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Idéia: Tentar dividir n por um primo de cada vez, em ordem crescente. Para cada primo visitado, divida o resultado obtido até falhar.

Considere que:

- `Primos` é uma lista inicializada com todos os primos conhecidos;
- `Resultado` é uma lista inicialmente vazia que terá a fatoração da entrada ao final.

Algoritmo: Fatoração de n

1. Para k em `Primos` faça $\{$
2.   Enquanto $n > 1$ e $k \mid n$ faça $\{$
3.   Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. $\}$
6.   Se $n = 1$, encerre.
7. $\}$

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 1 (“Para”):
Variáveis: $k = 2$, $n = 21$.

Linha 2. Como $2 \nmid 21$, não entramos no “Enquanto”.

$$\begin{array}{r|l} 21 & \\ & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 1 ("Para"):
Variáveis: $k = 2$, $n = 21$.

Linha 2. Como $2 \nmid 21$, não entramos no "Enquanto".
Linha 6. Como $21 \neq 1$, continuamos.

$21 \mid$

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 2 ("Para"):
Variáveis: $k = 3$, $n = 21$.

Linha 2. Como $3 \mid 21$, entramos no "Enquanto"

21

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 21$.

Linha 2. Como $3 \mid 21$, entramos no “Enquanto”
Linha 3. Escrevemos 3 em `Resultado`

$$\begin{array}{r|l} 21 & 3 \\ & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 2 ("Para"):
Variáveis: $k = 3$, $n = 7$.

Linha 2. Como $3 \mid 21$, entramos no "Enquanto"
Linha 3. Escrevemos 3 em `Resultado`
Linha 4. e atualizamos n para 21 **div** $3 = 7$.

21	3
7	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de *n*

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 7$.

Linha 2. Como $3 \nmid 7$, saímos do “Enquanto”

$$\begin{array}{r|l} 21 & 3 \\ 7 & \\ & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2.   Enquanto $n > 1$ e $k \mid n$ faça {
3.   Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5.   }
6.   Se $n = 1$, encerre.
7.   }

Executaremos o algoritmo para $n = 21$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 7$.

Linha 2. Como $3 \nmid 7$, saímos do “Enquanto”
Linha 6. Como $7 \neq 1$, continuamos.

$$\begin{array}{r|l} 21 & 3 \\ 7 & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 3 ("Para"):
Variáveis: $k = 5$, $n = 7$.

Linha 2. Como $5 \nmid 7$, não entramos no "Enquanto"

21	3
7	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 3 ("Para"):
Variáveis: $k = 5$, $n = 7$.

Linha 2. Como $5 \nmid 7$, não entramos no "Enquanto"
Linha 6. Como $7 \neq 1$, continuamos.

$$\begin{array}{r|l} 21 & 3 \\ 7 & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 4 (“Para”):
Variáveis: $k = 7$, $n = 7$.

Linha 2. Como $7 \mid 7$, não entramos no “Enquanto”

21	3
7	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 4 ("Para"):
Variáveis: $k = 7$, $n = 7$.

Linha 2. Como $7 \mid 7$, não entramos no "Enquanto"
Linha 3. Escrevemos 7 em `Resultado`

$$\begin{array}{r|l} 21 & 3 \\ 7 & 7 \\ & \end{array}$$

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 4 ("Para"):
Variáveis: $k = 7$, $n = 1$.

Linha 2. Como $7 \mid 7$, não entramos no "Enquanto"
Linha 3. Escrevemos 7 em `Resultado`
Linha 4. e atualizamos n para 7 **div** $7 = 1$.

21	3
7	7
1	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 4 ("Para"):
Variáveis: $k = 7$, $n = 1$.

Linha 2. Como $7 \nmid 1$, saímos do "Enquanto"

$$\begin{array}{r|l} 21 & 3 \\ 7 & 7 \\ 1 & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n

1. Para k em `Primos` faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }

Executaremos o algoritmo para $n = 21$.

Iteração 4 ("Para"):
Variáveis: $k = 7$, $n = 1$.

Linha 2. Como $7 \nmid 1$, saímos do "Enquanto"
Linha 6. E agora, como $1 = 1$, encerramos.

$$\begin{array}{r|l} 21 & 3 \\ 7 & 7 \\ \hline 1 & 3.7 \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

É possível sermos mais eficientes.

- Pelo Teorema "primo até $\sqrt{n}$", podemos parar assim que $k > \sqrt{n}$ mesmo considerando atualizações de n.
- Isso nos permitiria inicializar `Primos` somente com os primos até $\sqrt{n}$.
- Nesse caso, quando $k > \sqrt{n}$, teremos $n = 1$ ou que n é primo.

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, **sendo $k \leq \sqrt{n}$**, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. **Se $n \neq 1$, escreva n no fim de `Resultado` e encerre.**

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 1 (“Para”):
Variáveis: $k = 2$, $n = 99$ $(k \leq \sqrt{n})$

Linha 2. Como $2 \nmid 99$, não entramos no “Enquanto”

$$\begin{array}{r|l} 99 & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 1 ("Para"):
Variáveis: $k = 2$, $n = 99$ $(k \leq \sqrt{n})$

Linha 2. Como $2 \nmid 99$, não entramos no "Enquanto"
Linha 6. Como $99 \neq 1$, continuamos.

99	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 99$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \mid 99$, entramos no “Enquanto”

99	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 99$ $(k \leq \sqrt{n})$

Linha 2. Como $3 \mid 99$, entramos no “Enquanto”
Linha 3. Escrevemos 3 em `Resultado`

$$\begin{array}{r|l} 99 & 3 \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça $\{$
2. Enquanto $n > 1$ e $k \mid n$ faça $\{$
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. $\}$
6. Se $n = 1$, encerre.
7. $\}$
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 33$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \mid 99$, entramos no “Enquanto”
Linha 3. Escrevemos 3 em `Resultado`
Linha 4. e atualizamos n para 99 **div** $3 = 33$.

99	3
33	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 33$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \mid 33$, continuamos no “Enquanto”

99	3
33	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 ("Para"):
Variáveis: $k = 3$, $n = 33$ $(k \leq \sqrt{n})$

Linha 2. Como $3 \mid 33$, continuamos no "Enquanto"
Linha 3. Escrevemos 3 em `Resultado`

99	3
33	3

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 11$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \mid 33$, continuamos no “Enquanto”
Linha 3. Escrevemos 3 em `Resultado`
Linha 4. e atualizamos n para 33 **div** $3 = 11$.

99	3
33	3
11	

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2.   Enquanto $n > 1$ e $k \mid n$ faça {
3.   Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5.   }
6.   Se $n = 1$, encerre.
7.   }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 11$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \nmid 11$, saímos do “Enquanto”

99	3
33	3
11	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2.   Enquanto $n > 1$ e $k \mid n$ faça {
3.   Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5.   }
6.   Se $n = 1$, encerre.
7.   }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 2 (“Para”):
Variáveis: $k = 3$, $n = 11$ $\quad (k \leq \sqrt{n})$

Linha 2. Como $3 \nmid 11$, saímos do “Enquanto”
Linha 6. Como $11 \neq 1$, continuamos.

$$\begin{array}{r|l} 99 & 3 \\ 33 & 3 \\ 11 & \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 3 ("Para"):
Variáveis: $k = 5$, $n = 11$ **($k > \sqrt{n}$)**

Linha 1. Como $5 > \sqrt{11}$, saímos do "Para"

99	3
33	3
11	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 3 ("Para"):
Variáveis: $k = 5$, $n = 11$ ($k > \sqrt{n}$)

Linha 1. Como $5 > \sqrt{11}$, saímos do "Para"
Linha 8. Como $11 \neq 1$,
escrevemos 11 em `Resultado`

99	3
33	3
11	11

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 3 (“Para”):
Variáveis: $k = 5$, $n = 11$ ($k > \sqrt{n}$)

Linha 1. Como $5 > \sqrt{11}$, saímos do “Para”
Linha 8. Como $11 \neq 1$,
escrevemos 11 em `Resultado`,
atualizamos n para 1

99	3
33	3
11	11
1	

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

O Algoritmo de Fatoração

Algoritmo: Fatoração de n (Revisado)

1. Para k em `Primos`, sendo $k \leq \sqrt{n}$, faça {
2. Enquanto $n > 1$ e $k \mid n$ faça {
3. Escreva k no fim de `Resultado`
4. $n := n$ **div** k
5. }
6. Se $n = 1$, encerre.
7. }
8. Se $n \neq 1$, escreva n no fim de `Resultado`, $n := 1$, e encerre.

Executaremos o algoritmo para $n = 99$.

Iteração 3 ("Para"):
Variáveis: $k = 5$, $n = 11$ $(k > \sqrt{n})$

Linha 1. Como $5 > \sqrt{11}$, saímos do "Para"
Linha 8. Como $11 \neq 1$,
escrevemos 11 em `Resultado`,
atualizamos n para 1, e encerramos.

$$\begin{array}{r|l} 99 & 3 \\ 33 & 3 \\ 11 & 11 \\ \hline 1 & 3^2.11 \end{array}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicações da Fatoração de Inteiros

Pense na fatoração de um inteiro $n > 1$ como sendo o DNA deste número.

- Números diferentes terão fatorações diferentes
- Podemos representar números muito grandes usando números relativamente pequenos

Exemplos

- $2^{10} = 1024$ *(4 dígitos)*
- $2^{20} = 1048576$ *(7 dígitos)*
- $2^{30} = 1073741824$ *(10 dígitos)*
- $2^{10}.3^{10} = 60466176$ *(8 dígitos)*
- $2^{10}.5^{10} = 10000000000$ *(11 dígitos)*
- $3^{10} = 59049$ *(5 dígitos)*
- $3^{20} = 3486784401$ *(10 dígitos)*
- $3^{30} = 2{,}05891132095E{+}14$ *(15 dígitos)*
- $3^{10}.5^{10} = 576650390625$ *(12 dígitos)*
- $3^{10}.5^{10}.7^{10} = 1{,}62889462678E{+}20$ *(21 dígitos)*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicações da Fatoração de Inteiros

Pense na fatoração de um inteiro $n > 1$ como sendo o DNA deste número.

- Números diferentes terão fatorações diferentes
- Podemos representar números muito grandes usando números relativamente pequenos
- Todos os divisores de n estarão codificados

Exemplo

Calculamos anteriormente que $99 = 3^2.11$

Note que

1. *Primos cujo expoente não foram anotados têm expoente igual a* 1*.*
2. *Primos que não foram anotados têm expoente igual a* 0*.*

Ou seja*, calculamos* $99 = 3^2.11^1 = 2^0.3^2.5^0.7^0.11^1.13^0.17^0.\cdots$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicações da Fatoração de Inteiros

Pense na fatoração de um inteiro $n > 1$ como sendo o DNA deste número.

- Números diferentes terão fatorações diferentes
- Podemos representar números muito grandes usando números relativamente pequenos
- Todos os divisores de *n* estarão codificados

Exemplo

Calculamos anteriormente que $99 = 3^2.11^1$

Do ponto de vista de que $99 = 2^0.3^2.5^0.7^0.11^1.13^0.17^0.\cdots$

- *Não existe nenhum k múltiplo de* 2 *tal que* $k \mid 99$,
- $3 \mid 99$ ***(!)***,
- *Não existe nenhum k múltiplo de* 5 *tal que* $k \mid 99$,
- *Não existe nenhum k múltiplo de* 7 *tal que* $k \mid 99$,
- $11 \mid 99$ ***(!)***,
- *Não existe nenhum k múltiplo de* 13 *tal que* $k \mid 99$,
- *Não existe nenhum k múltiplo de* 17 *tal que* $k \mid 99$,
- ...

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Encontrar Divisores

Pense na fatoração de um inteiro $n > 1$ como sendo o DNA deste número.

- Números diferentes terão fatorações diferentes
- Podemos representar números muito grandes usando números relativamente pequenos
- Todos os divisores de n estarão codificados

Exemplo

Calculamos anteriormente que $99 = 3^2.11^1$

Todos *os divisores de* 99 *estão codificados na sua forma fatorada.*

- $3^0.11^0 = 1$
- $3^0.11^1 = 11$
- $3^1.11^0 = 3$
- $3^1.11^1 = 33$
- $3^2.11^0 = 9$
- $3^2.11^1 = 99$

Por quê? *Independente de como escrevemos* 99*, seus divisores serão os mesmos, mas a forma fatorada revela também os primos que podem estes divisores.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Encontrar Divisores

Pense na fatoração de um inteiro $n > 1$ como sendo o DNA deste número.

- Números diferentes terão fatorações diferentes
- Podemos representar números muito grandes usando números relativamente pequenos
- Todos os divisores de n estarão codificados

Exemplo

Calculamos anteriormente que $99 = 3^2.11^1$

Todos *os divisores de* 99 *estão codificados na sua forma fatorada.*

- $3^0.11^0 = 1$
- $3^1.11^0 = 3$
- $3^2.11^0 = 9$
- $3^0.11^1 = 11$
- $3^1.11^1 = 33$
- $3^2.11^1 = 99$

Então *basta variarmos os expoentes de cada primo de* 0 *até o valor de expoente que esse primo apresenta na fatoração de n para encontrarmos todos os divisores de n.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Calcular Quantidade de Divisores

E se quisermos saber apenas **quantos** divisores um número tem?

A fatoração do número também diz isso.

Exemplo

Calculamos anteriormente que $99 = 3^2.11^1$

E vimos que *basta variarmos os expoentes de cada primo de* 0 *até o valor de expoente que esse primo apresenta na fatoração de n para encontrarmos todos os divisores de n.*

Então...

- *se variarmos o expoente de* 3 *de* 0 *até* 2*, teremos* $2 + 1 = 3$ *possíveis valores.*
- *se variarmos o expoente de* 11 *de* 0 *até* 1*, teremos* $1 + 1 = 2$ *possíveis valores.*

Como os expoentes são indepententes, devemos multiplicar os números de possíveis valores.

Concluímos*, portanto, que* 99 *tem* $(2 + 1).(1 + 1) = 3.2 = 6$ *divisores.*

Samy Sá (samy@ufc.br)

Aplicação: Calcular Quantidade de Divisores

E se quisermos saber apenas **quantos** divisores um número tem?

A fatoração do número também diz isso.

Exemplo

Pelo Algoritmo de Fatoração, encontraremos que $120 = 2^3.3^1.5^1$

Então...

- *se variarmos o expoente de* 2 *de* 0 *até* 3, *teremos* $3 + 1 = 4$ *possíveis valores.*
- *se variarmos o expoente de* 3 *de* 0 *até* 1, *teremos* $1 + 1 = 2$ *possíveis valores.*
- *se variarmos o expoente de* 5 *de* 0 *até* 1, *teremos* $1 + 1 = 2$ *possíveis valores.*

Como os expoentes são indepententes, devemos multiplicar os números de possíveis valores.

Concluímos *que* 120 *tem* $(3 + 1).(1 + 1).(1 + 1) = 4.2.2 = 16$ *divisores.*

Aplicação: Calcular Quantidade de Divisores

E se quisermos saber apenas **quantos** divisores um número tem?

A fatoração do número também diz isso.

Exemplo

Pelo Algoritmo de Fatoração, encontraremos que $120 = 2^3.3^1.5^1$

De fato, 120 ***tem*** 16 ***divisores...***

- $2^0.3^0.5^0 = 1$
- $2^0.3^0.5^1 = 5$
- $2^0.3^1.5^0 = 3$
- $2^0.3^1.5^1 = 15$
- $2^1.3^0.5^0 = 2$
- $2^1.3^0.5^1 = 10$
- $2^1.3^1.5^0 = 6$
- $2^1.3^1.5^1 = 30$
- $2^2.3^0.5^0 = 4$
- $2^2.3^0.5^1 = 20$
- $2^2.3^1.5^0 = 12$
- $2^2.3^1.5^1 = 60$
- $2^3.3^0.5^0 = 8$
- $2^3.3^0.5^1 = 40$
- $2^3.3^1.5^0 = 24$
- $2^3.3^1.5^1 = 120$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de s e t*** *é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Notação

A função $\texttt{MDC}(s, t)$ retorna o máximo divisor comum de s, t.

Como já sabemos calcular divisores de um número por fatoração, temos como calcular os divisores comuns a dois ou mais números...

... E se sabemos encontrar divisores comuns a dois ou mais números, podemos indentificar o maior deles!

Estratégia 1: Calcular os divisores dos dois números e compará-los.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de s e t*** *é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Exemplo

Pelo Algoritmo de Fatoração, encontramos que

$$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$$

Então calculamos que...

1. *os divisores de* 99 *são* $1, 3, 9, 11, 33, 99$
2. *os divisores de* 120 *são* $1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

*Dados dois inteiros s e t diferentes de zero, o **máximo divisor comum de s e t** é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Exemplo

Pelo Algoritmo de Fatoração, encontramos que
$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$

Então calculamos que...

1. *os divisores de* 99 *são* $1, 3, 9, 11, 33, 99$
2. *os divisores de* 120 *são* $1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120$

***Comparando as listas**, os únicos divisores comuns de* 99 *e* 120 *são* 1 *e* 3.
***Portanto**,* $\texttt{MDC}(99, 120) = 3$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de*** *s* ***e*** *t é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Notação

A função $\texttt{MDC}(s, t)$ retorna o máximo divisor comum de s, t.

Como já sabemos calcular divisores de um número por fatoração, temos como calcular os divisores comuns a dois ou mais números...

... E se sabemos encontrar divisores comuns a dois ou mais números, podemos indentificar o maior deles!

Estratégia 2: Calcular o MDC a partir dos **fatores comuns**.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de** s **e** t *é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Exemplo

Pelo Algoritmo de Fatoração, encontramos que

$$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$$

Note que...

- *Todo divisor de* 99 *é combinação de* $(3^0, 3^1, 3^2)$ *com* $(11^0, 11^1)$
- *Todo divisor de* 120 *é combinação de* $(2^0, 2^1, 2^2, 2^3)$ *com* $(3^0, 3^1)$ *e com* $(5^0, 5^1)$*.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de*** s ***e*** t *é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Exemplo

Pelo Algoritmo de Fatoração, encontramos que

$$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$$

Note que...

- *Todo divisor de* 99 *é combinação de* $(3^0, 3^1, 3^2)$ *com* $(11^0, 11^1)$
- *Todo divisor de* 120 *é combinação de* $(2^0, 2^1, 2^2, 2^3)$ *com* $(3^0, 3^1)$ *e com* $(5^0, 5^1)$

Então *os divisores comuns de* 99 *e* 120 ***só podem ser*** *combinações de* 3^0 *ou* 3^1*, o que nos permite construir apenas os números* 1 *e* 3*.* ***Portanto****,* $\texttt{MDC}(99, 120) = 3$*.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Definição

Dados dois inteiros s e t diferentes de zero, o ***máximo divisor comum de s e t*** *é o maior inteiro d tal que $d \mid s$ e $d \mid t$.*

Teorema

Dados s, t inteiros positivos, considere que $p_1, p_2, \cdots, p_n$ são todos os primos que ocorrem com expoentes positivos nas fatorações de s ou de t.

Isto nos permite escrever que

$$s = p_1^{a_1}.p_2^{a_2}.\cdots.p_n^{a_n}$$
$$t = p_1^{b_1}.p_2^{b_2}.\cdots.p_n^{b_n}$$

Então $\mathtt{MDC}(s,t) = p_1^{min(a_1,b_1)}.p_2^{min(a_2,b_2)}.\cdots.p_n^{min(a_n,b_n)}$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MDC

Teorema

Dados s, t inteiros positivos, considere que $p_1, p_2, \cdots, p_n$ são todos os primos que ocorrem com expoentes positivos nas fatorações de s ou de t.

Isto nos permite escrever que

$$s = p_1^{a_1}.p_2^{a_2}.\cdots.p_n^{a_n}$$
$$t = p_1^{b_1}.p_2^{b_2}.\cdots.p_n^{b_n}$$

Então $\mathtt{MDC}(s,t) = p_1^{min(a_1,b_1)}.p_2^{min(a_2,b_2)}.\cdots.p_n^{min(a_n,b_n)}$.

Exemplo

Pelo Algoritmo de Fatoração, encontramos que $99 = 3^2.11^1$ *e* $120 = 2^3.3^1.5^1$.

Os primos que ocorrem com expoentes positivos nestas fatorações são $2, 3, 5$ *e* 11*, então percebamos estas fatorações como* $99 = 2^0.3^2.5^0.11^1$ *e* $120 = 2^3.3^1.5^1.11^0$.

Então $\mathtt{MDC}(99,120) = 2^{min(0,3)}.3^{min(2,1)}.5^{min(0,1)}.11^{min(1,0)}$
$= 2^0.3^1.5^0.11^0 = 3$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Definição

Dados dois inteiros s e t diferentes de zero, o ***mínimo múltiplo comum de s e t*** *é o menor inteiro* ***positivo*** *d tal que $s \mid d$ e $t \mid d$.*

Notação

A função $\texttt{MMC}(s, t)$ retorna o mínimo múltiplo comum de s, t.

Para encontrar múltiplos de um número, basta multiplicá-lo pelos inteiros positivos...

... mas cada inteiro terá infinitos múltiplos positivos, dificultando encontrarmos os números que se repetem nas duas listas.

Estratégia 1: Calcular o MMC a partir dos **fatores comuns**.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Definição

Dados dois inteiros s e t diferentes de zero, o ***mínimo múltiplo comum de*** s ***e*** t *é o menor inteiro* ***positivo*** d *tal que* $s \mid d$ *e* $t \mid d$.

Exemplo

Pelo Algoritmo de Fatoração, encontramos que

$$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$$

Note que...

- *Todo múltiplo de* 99 *precisa reter os fatores* 3^2 *e* 11^1.
- *Todo múltiplo de* 120 *precisa reter os fatores* 2^3, 3^1 *e* 5^1.

Então *o menor múltiplo comum de* 99 *e* 120 *terá estes fatores e nada mais. Além disso, como* $3^1 \mid 3^2$, *basta usarmos* 3^2 *para incluir os dois fatores de base* 3.

Portanto, $\texttt{MMC}(99, 120) = 2^3.3^2.5^1.11^1 = 3960$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Definição

Dados dois inteiros s e t diferentes de zero, o ***mínimo múltiplo comum de*** *s* ***e*** *t é o menor inteiro* ***positivo*** *d tal que $s \mid d$ e $t \mid d$.*

Exemplo

Pelo Algoritmo de Fatoração, encontramos que

$$99 = 3^2.11^1 \quad e \quad 120 = 2^3.3^1.5^1$$

Observe que...

- 3960 ***div*** $99 = 40$
- 3960 ***div*** $120 = 33$

Ou seja, *a estratégia de calcular múltiplos de cada número para comparar as duas listas exigiria ao menos 40 múltiplos de 99 e 33 múltiplos de 120, o que não é nada eficiente.*

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Definição

Dados dois inteiros s e t diferentes de zero, o ***mínimo múltiplo comum de s e t*** *é o menor inteiro* ***positivo*** *d tal que $s \mid d$ e $t \mid d$.*

Teorema

Dados s, t inteiros positivos, considere que $p_1, p_2, \cdots, p_n$ são todos os primos que ocorrem com expoentes positivos nas fatorações de s ou de t.

Isto nos permite escrever que

$$s = p_1^{a_1}.p_2^{a_2}.\cdots.p_n^{a_n}$$
$$t = p_1^{b_1}.p_2^{b_2}.\cdots.p_n^{b_n}$$

Então $\mathtt{MMC}(s,t) = p_1^{max(a_1,b_1)}.p_2^{max(a_2,b_2)}.\cdots.p_n^{max(a_n,b_n)}$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, considere que $p_1, p_2, \cdots, p_n$ são todos os primos que ocorrem com expoentes positivos nas fatorações de s ou de t.

Isto nos permite escrever que

$$s = p_1^{a_1}.p_2^{a_2}.\cdots.p_n^{a_n}$$
$$t = p_1^{b_1}.p_2^{b_2}.\cdots.p_n^{b_n}$$

Então $\mathtt{MMC}(s,t) = p_1^{max(a_1,b_1)}.p_2^{max(a_2,b_2)}.\cdots.p_n^{max(a_n,b_n)}$.

Exemplo

Pelo Algoritmo de Fatoração, encontramos que $99 = 3^2.11^1$ *e* $120 = 2^3.3^1.5^1$.

Os primos que ocorrem com expoentes positivos nestas fatorações são $2, 3, 5$ *e* 11*, então convém entendermos estas fatorações como* $99 = 2^0.3^2.5^0.11^1$ *e* $120 = 2^3.3^1.5^1.11^0$.

Então $\mathtt{MMC}(99, 120) = 2^{max(0,3)}.3^{max(2,1)}.5^{max(0,1)}.11^{max(1,0)}$
$= 2^3.3^2.5^1.11^1 = 3960$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Definição

Dados dois inteiros s e t diferentes de zero, o ***mínimo múltiplo comum de s e t*** *é o menor inteiro* ***positivo*** *d tal que $s \mid d$ e $t \mid d$.*

Notação

A função $\texttt{MMC}(s, t)$ retorna o mínimo múltiplo comum de s, t.

Para encontrar múltiplos de um número, basta multiplicá-lo pelos inteiros positivos...

... mas cada inteiro terá infinitos múltiplos positivos, dificultando encontrarmos os números que se repetem nas duas listas.

Estratégia 2: Calcular o MMC a partir do MDC.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\mathtt{MMC}(s,t).\mathtt{MDC}(s,t) = s.t$.

Exemplo

Anteriormente, calculamos $\mathtt{MMC}(99,120) = 3960$ *e* $\mathtt{MDC}(99,120) = 3$.

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Ou seja, $\mathtt{MMC}(99,120).\mathtt{MDC}(99,120) = 99.120$, como diz o teorema.

Então, se já soubermos que $\mathtt{MDC}(99,120) = 3$, podemos calcular $\mathtt{MMC}(99,120)$, pois o teorema nos dirá que $\mathtt{MMC}(99,120).3 = 99.120$,
ou seja, $\mathtt{MMC}(99,120) = \frac{\cancel{9}9.120}{\cancel{3}} = 33.120 = 3960$.

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\mathtt{MMC}(s,t).\mathtt{MDC}(s,t) = s.t$*.*

Exemplo

Anteriormente, calculamos $\mathtt{MMC}(99,120) = 3960$ *e* $\mathtt{MDC}(99,120) = 3$*.*

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Mas por que isso acontece? (1/2)

Lembre que $99 = 2^0.3^2.5^0.11^1$ e $120 = 2^3.3^1.5^1.11^0$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t *inteiros positivos, teremos que* $\texttt{MMC}(s,t).\texttt{MDC}(s,t) = s.t$.

Exemplo

Anteriormente, calculamos $\texttt{MMC}(99, 120) = 3960$ *e* $\texttt{MDC}(99, 120) = 3$.

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Mas por que isso acontece? (1/2)

Lembre que $99 = 2^0.3^2.5^0.11^1$ e $120 = 2^3.3^1.5^1.11^0$

Daí, calculamos

$\texttt{MDC}(99, 120) = 2^{min(0,3)}.3^{min(2,1)}.5^{min(0,1)}.11^{min(1,0)}$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\texttt{MMC}(s,t).\texttt{MDC}(s,t) = s.t$.

Exemplo

Anteriormente, calculamos $\texttt{MMC}(99,120) = 3960$ *e* $\texttt{MDC}(99,120) = 3$.

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Mas por que isso acontece? (1/2)

Lembre que $99 = 2^0.3^2.5^0.11^1$ e $120 = 2^3.3^1.5^1.11^0$

Daí, calculamos

$$\texttt{MDC}(99,120) = 2^{min(0,3)}.3^{min(2,1)}.5^{min(0,1)}.11^{min(1,0)} = 2^0.3^1.5^0.11^0$$

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\mathtt{MMC}(s, t).\mathtt{MDC}(s, t) = s.t$.

Exemplo

Anteriormente, calculamos $\mathtt{MMC}(99, 120) = 3960$ *e* $\mathtt{MDC}(99, 120) = 3$.

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Mas por que isso acontece? (1/2)

Lembre que $99 = 2^0.3^2.5^0.11^1$ e $120 = 2^3.3^1.5^1.11^0$

Daí, calculamos

$\mathtt{MDC}(99, 120) = 2^{min(0,3)}.3^{min(2,1)}.5^{min(0,1)}.11^{min(1,0)} = 2^0.3^1.5^0.11^0$

e $\mathtt{MMC}(99, 120) = 2^{max(0,3)}.3^{max(2,1)}.5^{max(0,1)}.11^{max(1,0)}$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\texttt{MMC}(s, t).\texttt{MDC}(s, t) = s.t$.

Exemplo

Anteriormente, calculamos $\texttt{MMC}(99, 120) = 3960$ *e* $\texttt{MDC}(99, 120) = 3$.

Note que...

- $3960.3 = 11880$
- $99.120 = 11880$

Mas por que isso acontece? (1/2)

Lembre que $99 = 2^0.3^2.5^0.11^1$ e $120 = 2^3.3^1.5^1.11^0$

Daí, calculamos

$\texttt{MDC}(99, 120) = 2^{min(0,3)}.3^{min(2,1)}.5^{min(0,1)}.11^{min(1,0)} = 2^0.3^1.5^0.11^0$

e $\texttt{MMC}(99, 120) = 2^{max(0,3)}.3^{max(2,1)}.5^{max(0,1)}.11^{max(1,0)} = 2^3.3^2.5^1.11^1$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Aplicação: Cálculo de MMC

Teorema

Dados s, t inteiros positivos, teremos que $\texttt{MMC}(s,t).\texttt{MDC}(s,t) = s.t.$

Mas por que isso acontece? (2/2)

Por outra perspectiva, nossos números são

$$99 = 2^0.3^2.5^0.11^1 \qquad \texttt{MDC}(99,120) = 2^0.3^1.5^0.11^0$$
$$120 = 2^3.3^1.5^1.11^0 \qquad \texttt{MMC}(99,120) = 2^3.3^2.5^1.11^1$$

Então,

$$\begin{aligned} 99.120 &= (2^0.3^2.5^0.11^1).(2^3.3^1.5^1.11^0) \\ &= 2^0.3^2.5^0.11^1 \;.\; 2^3.3^1.5^1.11^0 \\ &= 2^3.3^2.5^1.11^1 \;.\; 2^0.3^1.5^0.11^0 \\ &= (2^3.3^2.5^1.11^1).(2^0.3^1.5^0.11^0) \\ &= \texttt{MMC}(99,120) \;.\; \texttt{MDC}(99,120) \end{aligned}$$

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Roteiro

Universidade Federal do Ceará
Computer Science Department
Samy Sá (samy@ufc.br)

Divisores Comuns

Teorema

Sejam s, t inteiros positivos, então $d \mid s$ e $d \mid t$ se e somente se $d \mid \texttt{MDC}(s,t)$.

Os divisores comuns a s, t são exatamente os divisores do seu MDC.

Exemplo

Para encontrar os divisores comuns de 120 *e* 75*?*

Basta calcular os divisores de $\texttt{MDC}(120, 75)$.

Com os métodos discutidos nesta apresentação,

1. *Fatoramos* 120 *e* 75*, obtendo que* $75 = 3^1.5^2$ *e* $120 = 2^3.3^1.5^1$.
2. *Calculamos* $\texttt{MDC}(120,75) = 2^{min(0,3)}.3^{min(1,1)}.5^{min(2,1)} = 2^0.3^1.5^1 = 15$
3. *Calculamos os divisores positivos de* 15

$$3^0.5^0 = 1 \qquad 3^0.5^1 = 5 \qquad 3^1.5^0 = 3 \qquad 3^1.5^1 = 15$$

Então *os divisores (positivos) comuns de* 120 *e* 75 *são* 1, 3, 5 *e* 15.