

Máximo Divisor Comum - MDC

Prof. Antonio Carlos Mariani

INE | CTC



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Problema

Dados dois números naturais quaisquer, desejam calcular o MDC (Máximo divisor comum) destes números.

Por exemplo, se tomarmos os número 16 e 24, o maior número que os divide simultaneamente é 8. Dizemos, portanto, que

$$\text{MDC}(16, 24) = 8$$

De forma similares, se tomarmos os números 30, 50 e 20, vemos que o máximo divisor comum é 10.

$$\text{MDC}(30, 50, 20) = 10$$

Como calcular o MDC?

Cálculo do MDC

O método mais clássico de cálculo do MDC envolve três passos:

1. Decompor os números dados em fatores primos;
2. Pegar os fatores primos comuns;
3. Fazer o produtos desses fatores.

Vamos encontrar o máximo divisor comum para os números 80 e 1800.

$$\bullet 80 = \overset{\text{red}}{\circledast} 2 \times \overset{\text{blue}}{\circledast} 2 \times \overset{\text{green}}{\circledast} 2 \times 2 \times \overset{\text{purple}}{\circledast} 5$$

$$\bullet 1800 = \overset{\text{red}}{\circledast} 2 \times \overset{\text{blue}}{\circledast} 2 \times \overset{\text{green}}{\circledast} 2 \times 3 \times 3 \times \overset{\text{purple}}{\circledast} 5 \times 5$$

$$\text{Logo, } \text{MDC}(80, 1800) = \overset{\text{red}}{\circledast} 2 \times \overset{\text{blue}}{\circledast} 2 \times \overset{\text{green}}{\circledast} 2 \times \overset{\text{purple}}{\circledast} 5 = 40$$

Decomposição simultânea

Série de iterações: cada linha da tabela

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

Fatores primos que dividem simultaneamente os dois valores

$$\text{MDC}(80, 1800) = 2 \times 2 \times 2 \times 5 = 40$$

Condição de parada

Decomposição simultânea

Obtenção dos dados de entrada e impressão do resultado

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×  
1 n1 = int(input())  
2 n2 = int(input())  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17 print(mdc)  
18
```

Decomposição simultânea

Repetição (iterações) com base numa condição

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4
5
6  while n1 ou n2 > 1:
7
8
9
10
11
12
13
14
15
16
17
18  print(mdc)
```

Condição de parada: n1 e n2 iguais a 1

Decomposição simultânea

O que se repete em cada iteração?

	n1	n2	primo
1 ^a	80	1800	2
2 ^a	40	900	2
3 ^a	20	450	2
4 ^a	10	225	2
5 ^a	5	225	2 → 3
6 ^a	5	75	3
7 ^a	5	25	3 → 5
8 ^a	1	5	5
9 ^a	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4
5
6  while n1 ou n2 > 1:
7
8
9
10
11
12
13
14
15
16
17
18  print(mdc)
```



Decomposição simultânea

Primo inicia com 2 e segundo alguma condição passa para o próximo

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4
5  primo = 2
6  while n1 ou n2 > 1:
7
8
9
10
11
12
13
14
15
16  primo = número primo seguinte ao atual
17
18  print(mdc)
```

Decomposição simultânea

Procurando padrões nas iterações

	n1	n2	primo
1 ^a	80	1800	2
2 ^a	40	900	2
3 ^a	20	450	2
4 ^a	10	225	2
5 ^a	5	225	2 → 3
6 ^a	5	75	3
7 ^a	5	25	3 → 5
8 ^a	1	5	5
9 ^a	1	1	

Quatro situações e comportamentos distintos:

1. Tanto n1 como n2 são divisíveis pelo primo:

2. Apenas n1 é divisível pelo primo

3. Apenas n2 é divisível pelo primo

4. Nenhum dos valores é divisível pelo primo

Vamos analisar cada um dos casos.

Decomposição simultânea

1º) os dois valores são divisíveis: divide-se

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4
5  primo = 2
6  while n1 ou n2 > 1:
7      if n1 e n2 são divisíveis por primo:
8
9          n1 //= primo # divisão inteira
10         n2 //= primo # divisão inteira
11
12
13
14
15
16         primo = número primo seguinte ao atual
17
18  print(mdc)
```

Decomposição simultânea

1º) acumula-se progressivamente o produto

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  mdc = 1
5  primo = 2
6  while n1 ou n2 > 1:
7      if n1 e n2 são divisíveis por primo:
8          mdc *= primo # primo nos interessa para cálculo do MDC
9          n1 //= primo # divisão inteira
10         n2 //= primo # divisão inteira
11
12
13
14
15
16         primo = número primo seguinte ao atual
17
18  print(mdc)
```

$$\text{MDC}(80, 1800) = 2 \times 2 \times 2 \times 5 = 40$$

Decomposição simultânea

2º) e 3º) apenas um dos valores é divisível: divide-se

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  mdc = 1
5  primo = 2
6  while n1 ou n2 > 1:
7      if n1 e n2 são divisíveis por primo:
8          mdc *= primo # primo nos interessa para cálculo do MDC
9          n1 //= primo # divisão inteira
10         n2 //= primo # divisão inteira
11         elif n1 é divisível por primo:
12             n1 //= primo
13         elif n2 é divisível por primo:
14             n2 //= primo
15
16         primo = número primo seguinte ao atual
17
18  print(mdc)
```

Decomposição simultânea

4º) nenhum dos valores é divisível: próximo primo

	n1	n2	primo
1ª	80	1800	2
2ª	40	900	2
3ª	20	450	2
4ª	10	225	2
5ª	5	225	2 → 3
6ª	5	75	3
7ª	5	25	3 → 5
8ª	1	5	5
9ª	1	1	

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  mdc = 1
5  primo = 2
6  while n1 ou n2 > 1:
7      if n1 e n2 são divisíveis por primo:
8          mdc *= primo # primo nos interessa para cálculo do MDC
9          n1 //= primo # divisão inteira
10         n2 //= primo # divisão inteira
11     elif n1 é divisível por primo:
12         n1 //= primo
13     elif n2 é divisível por primo:
14         n2 //= primo
15     else:
16         primo = número primo seguinte ao atual
17
18  print(mdc)
```

Como são 4 os casos possíveis, não sendo nenhum dos 3 anteriores, só resta este (cláusula else)

Decomposição simultânea

Substituindo os textos em português por código Python

```
mdc.py ×
1 n1 = int(input())
2 n2 = int(input())
3
4 mdc = 1
5 primo = 2
6 while n1 ou n2 > 1:
7     if n1 e n2 são divisíveis por primo:
8         mdc *= primo # primo nos interessa para cálculo do MDC
9         n1 //= primo # divisão inteira
10        n2 //= primo # divisão inteira
11    elif n1 é divisível por primo:
12        n1 //= primo
13    elif n2 é divisível por primo:
14        n2 //= primo
15    else:
16        primo = número primo seguinte ao atual
17
18 print(mdc)
```

Decomposição simultânea

Escrever rotina que obtenha o próximo primo (fica como tarefa)

```
mdc.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  mdc = 1
5  primo = 2
6  while n1 > 1 or n2 > 1:
7      if n1 % primo == 0 and n2 % primo == 0:
8          mdc *= primo # primo nos interessa para cálculo do MDC
9          n1 //= primo # divisão inteira
10         n2 //= primo # divisão inteira
11     elif n1 % primo == 0:
12         n1 //= primo
13     elif n2 % primo == 0:
14         n2 //= primo
15     else:
16         primo = número primo seguinte ao atual
17
18  print(mdc)
```

Cálculo do MDC

Outra abordagem?

Perguntas:

- O método de decomposição em fatores primos é o único possível?
- Será que ele é o melhor (mais eficaz)?

Uma coisa é certa: quem não procura, não acha!

Divisões sucessivas

Tentativas de divisões sucessivas e concomitantes pela sequência dos números naturais iniciando em 2: [2, 3, 4, 5, 6, ...]

1º estimativa: o nº 1 sempre divide os dois valores

$$\begin{array}{r} n1 \\ \hline 16 \end{array} \quad \begin{array}{r} n2 \\ \hline 24 \end{array}$$

divisor

1 → ok

2 → ok

3

4 → ok

5

6

7

8 → ok

9

10

...

?

Progressivamente procura-se novas estimativas

- Até onde deveríamos ir?
- Como saber que encontramos o MDC e, portanto, podemos parar a busca?

Resposta: como o MDC divide simultaneamente os dois valores $n1$ e $n2$, então ele não pode ser maior que o menor dos dois valores, ou seja:

- $\text{mdc}(n1, n2) \leq \min(n1, n2)$

Neste exemplo: $\text{mdc}(16, 24) \leq 16$

Cálculo do MDC

Tentativas de divisões sucessivas e concomitantes pela sequência dos números naturais iniciando em 2: [2, 3, 4, 5, 6, ...]

divisor

1 → ok
2 → ok
3
4 → ok
5
6
7
8 → ok
9
10
...
16

$\frac{n1}{16}$ $\frac{n2}{24}$

Estimativa inicial para o mdc

```
mdc.py x
1  n1 = int(input())
2  n2 = int(input())
3
4  menor = min(n1, n2)
5  divisor = 2
6  mdc = 1
7  while divisor <= menor:
8      if n1 % divisor == 0 and n2 % divisor == 0:
9          mdc = divisor
10         divisor += 1
11
12  print(mdc)
```

Este método resulta num algoritmo mais simples, não sendo necessário conhecer a sequência de números primos.

Cálculo do MDC

- Mas será que não há um método mais eficiente?
- Como saber?

Para este caso, sim, o método de Euclides (por volta de 300 a.C.)

Método de Euclides

$$\begin{array}{r} 20 \overline{) 12} \\ \underline{-12} \\ 08 \end{array}$$
$$\begin{array}{r} 12 \overline{) 8} \\ \underline{-8} \\ 04 \end{array}$$
$$\begin{array}{r} 8 \overline{) 4} \\ \underline{-8} \\ 0 \end{array}$$

$$\text{MDC}(20, 12) = ?$$

- Divisões sucessivas
- Para quando resto = 0
- O último divisor é o MDC

$$\text{MDC}(20, 12) = 4$$

<https://brasilecola.uol.com.br/matematica>

Método de Euclides

n1	n2		
Dividendo	Divisor	Quociente	Resto
1128	336	3	120

Valores iniciais

Método de Euclides

n1	n2		
Dividendo	Divisor	Quociente	Resto
1128	336	3	120
336	120	2	96

The diagram illustrates the Euclidean algorithm steps. Blue arrows show the progression of values: 336 from the divisor of the first step to the dividend of the second, and 120 from the remainder of the first step to the divisor of the second. A green arrow shows the final remainder 96. A green box highlights the final remainder 96.

Método de Euclides

n1	n2		
Dividendo	Divisor	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24

The diagram illustrates the steps of the Euclidean algorithm. Blue arrows show the progression of the divisor from 336 to 120 to 96. A green bracket and arrow highlight the final remainder of 24.

Método de Euclides

n1	n2		
Dividendo	Divisor	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

Atingida a
condição de parada

Método de Euclides

n1	n2		
Dividendo	Divisor	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

$\text{MDC}(1128, 336) = 24$

Método de Euclides

n1	n2	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

```
mdc_euclides.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  resto = n1 % n2
5  while resto > 0:
6
7
8
9
10
```

Condição de parada

Método de Euclides

n1	n2	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

```
mdc_euclides.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  resto = n1 % n2
5  while resto > 0:
6      n1 = n2
7      n2 = resto
8      resto = n1 % n2
9
10
```

Método de Euclides

n1	n2	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

```
mdc_euclides.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  resto = n1 % n2
5  while resto > 0:
6      n1 = n2
7      n2 = resto
8      resto = n1 % n2
9
10
```

Método de Euclides

n1	n2	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

```
mdc_euclides.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  resto = n1 % n2
5  while resto > 0:
6      n1 = n2
7      n2 = resto
8      resto = n1 % n2
9
10
```

Método de Euclides

n1	n2	Quociente	Resto
1128	336	3	120
336	120	2	96
120	96	1	24
96	24	4	0

```
mdc_euclides.py ×
1  n1 = int(input())
2  n2 = int(input())
3
4  resto = n1 % n2
5  while resto > 0:
6      n1 = n2
7      n2 = resto
8      resto = n1 % n2
9
10 print(n2)
```

$$\text{MDC}(1128, 336) = 24$$

Contato

antonio.c.mariani@ufsc.br

CTC/INE



UNIVERSIDADE FEDERAL
DE SANTA CATARINA