

第1.3节 素数和最大公约数

Section 1.3: Primes and Greatest Common Divisors

知识要点

1

素数及其性质

2

素数的猜想和开放问题

3

最大公约数和最小公倍数

4

欧几里得算法

5

gcd的线性组合的表示

1.3.1 素数

- 【定义】：大于1的整数 p 称为**素数**(也叫质数), 如果 p 的正因子只有1和 p . 大于1但又不是素数的正整数集合叫做**合数**.
- 例如 $\{2, 3, 5, 7, 11, 13, 17, 19, 23, \dots\}$ 是素数集合. 1既不是素数, 也不是合数. 9是合数, 因为3整除9.

历史与重要性

- 研究可追溯至古希腊时期, 欧几里得证明了素数有无限多个;
- 素数是数论的基础, 被称为"数的原子", 任何大于1的自然数都可以唯一分解为素数的乘积;
- 现代密码学(如RSA加密)的安全性基于大素数分解的困难性.

【基础知识：假如整数 a 除以 b 等于一个没有余数的整数(也就是 $a|b$), 那么我们称 b 是 a 的**整数因子**. 比如 $42=6*7$, 因此7是42的因子. 正整数因子简称**正因子**或称**正因数**】

□ 【素数性质】

- 1、设 p 是素数, 且 $d|p$, 若 $d > 1$, 则 $d = p$
- 2、设 p 是素数, 且 $p|ab$, 则必有 $p|a$ 或者 $p|b$
- 3、整数 p 是合数当且仅当存在整数 a , 使得 $a|p$ 并且 $1 < a < p$
- 4、合数必有素数因子, 即设 a 是合数, 则存在素数 p , 使得 $p|a$

□ 证明略.

□ 根据性质4, 任何大于1的整数要么是素数, 要么可以分解成素数的乘积. 这就是下述的**算术基本定理**. 它表明素数是构成整数的基本元素.

1.3.1 素数

- **【算术基本定理】**：每个大于1的整数都可以唯一地写成两个或者多个素数的乘积, 其中素数因子(简称素因子)以非递减序排列.
- 备注:该定理的证明自行验证, 该定理主要针对合数.
- 例:分别求3, 1024, 7007这些数的素因子分解式
 - $3 = 3$
 - $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$
 - $7007 = 7^2 * 11 * 13$
 - 备注:素因子分解成这样的过程怎么实现呢?

□ 一个整数 n 的素因子分解方式:

- 从最小的素数2开始, 依次用这个素数去除 n . 如果不能2整除 n , 那么继续下一个更大的素数.
- 直到找到一个素因子 p . 那么, 继续对整除后的商 n/p 做素因子分解. 此时, 继续用当前的素数 p 进行除法操作, 也就是 n/p 除以 p 看结果是否有余数, 找到下一个素因子继续以上操作.
- 直到最后的商也是素数为止.

□ 例如对15进行素因子分解. 首先 $2 \nmid 15$, 接着分析 $3 \mid 15$. 因为 $15 = 3 * 5$, 商为5, 5也是一个素数, 因此 $15 = 3 * 5$ 就是最后的素因子分解式.

1.3.1 素数

□例:找出147的素因子分解式

□解:

- 不断地用素数去除147. 首先用2去除147, 不能整除.
- 素数2后面接着下一个素数是3. $3|147$, 且商为 $147/3=49$.
- 为此, 继续用3不断地除上一次的商49, 但不能整除.
- 素数3后面的下一个素数是5, 49除以5有余数.
- 继续找下一个素数7, $7|49$, 商为 $49/7=7$. 注意到商7也是一个素数, 分解过程完成.
- 因此 $147=3*49=3*7*7=3*7^2$

1.3.1 素数

□例:147的正因子个数是多少?(备注, 前面例题已求得 $147=3 * 7^2$)

□解:

➤所以正因子个数为 $(1+1)*(2+1)=6$

1.3.2 试除法

- 如何证明或者验证一个给定的整数 n 是素数(叫做**素数测试**)? 比如, 在密码学中大素数就用于为信息加密的某些方法中.
- (方法一):根据素数的定义, 我们可以从1开始, 一直到 n 循环验证这些整数能否整除 n . 如果 n 不能被除1和它自己以外其他的整数整除的话, 那么我们可以判定它是素数.
 - 备注:这是最蛮力的一一试除法, 效率低下.

1.3.2 试除法

- 【定理2】 如果 n 为一个合数, 那么 n 必有一个素因子小于或等于 \sqrt{n} .
- 该定理的证明自行验证.
- (方法二): 从以上定理可知, 如果一个整数不能被小于或等于其平方根的素数整除, 则它就是素数. 因此, 把 n 除以所有不超过 \sqrt{n} 的素数, 如果不能被其中任意一个素数整除, 则 n 为素数.

1.3.2 试除法

□例:证明101是素数

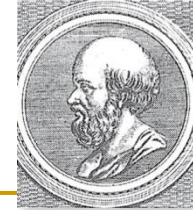
□解:不超过 $\sqrt{101}$ 的素数只有2, 3, 5, 7. 因为101不能被2, 3, 5, 7整除(101除以这些数都会有余数), 所以101是素数.

□例:证明4969是素数

□解:不超过 $\sqrt{4969} \approx 70.49$ 的素数有2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67. 因为4969不能被这些数整除(4969除以这些数都会有余数), 所以4969是素数.

➤备注:其中小于70.49的素数有哪些呢? 厄拉多塞筛法可以实现.

1.3.3 埃拉托斯特尼筛法



Eratosthenes
(276-194 B.C.)



华中科技大学
计算机科学与技术学院
School of Computer Science & Technology, HUST

- **埃拉托斯特尼筛法**(也称为**厄拉多塞筛法**, **Eratosthene筛法**)用来寻找不超过一个给定整数的所有素数.
- 例:寻找不超过100的素数.
- 解:首先构造1到100的全部整数的列表, 然后
 - 注意, 不超过100的合数必定有一个不超过10的素因子(2, 3, 5, 7). 所以不超过100的素数就是这四个素数以及那些大于1且不超过100同时不能被2, 3, 5, 7之一整除的正整数.
 - 除2以外, 删除那些能被2整除的整数.
 - 除3以外, 删除那些能被3整除的整数.
 - 除5以外, 删除那些能被5整除的整数.
 - 除7以外, 删除那些能被7整除的整数.
 - 因为所有不超过100的合数都能被2, 3, 5或7整除, 所以除了1以外, 所有保留下来的整数都是素数. 综上所述, 不超过100的素数包括以下共计25个素数:
{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}

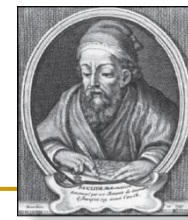
1.3.3 埃拉托斯特尼筛法

□ 图示:

TABLE 1 The Sieve of Eratosthenes.																			
Integers divisible by 2 other than 2 receive an underline.										Integers divisible by 3 other than 3 receive an underline.									
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>	1	2	3	<u>4</u>	5	<u>6</u>	7	8	<u>9</u>	<u>10</u>
11	<u>12</u>	13	<u>14</u>	15	<u>16</u>	17	<u>18</u>	19	<u>20</u>	11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>
21	<u>22</u>	23	<u>24</u>	25	<u>26</u>	27	<u>28</u>	29	<u>30</u>	<u>21</u>	<u>22</u>	23	<u>24</u>	25	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>
31	<u>32</u>	33	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>	31	<u>32</u>	<u>33</u>	<u>34</u>	35	<u>36</u>	37	<u>38</u>	<u>39</u>	<u>40</u>
41	<u>42</u>	43	<u>44</u>	45	<u>46</u>	47	<u>48</u>	49	<u>50</u>	41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>
51	<u>52</u>	53	<u>54</u>	55	<u>56</u>	57	<u>58</u>	59	<u>60</u>	<u>51</u>	<u>52</u>	53	<u>54</u>	55	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>
61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>	61	<u>62</u>	<u>63</u>	<u>64</u>	65	<u>66</u>	67	<u>68</u>	<u>69</u>	<u>70</u>
71	<u>72</u>	73	<u>74</u>	75	<u>76</u>	77	<u>78</u>	79	<u>80</u>	71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>
81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	87	<u>88</u>	89	<u>90</u>	<u>81</u>	<u>82</u>	83	<u>84</u>	85	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>
91	<u>92</u>	93	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>	91	<u>92</u>	<u>93</u>	<u>94</u>	95	<u>96</u>	97	<u>98</u>	<u>99</u>	<u>100</u>
Integers divisible by 5 other than 5 receive an underline.										Integers divisible by 7 other than 7 receive an underline; integers in color are prime.									
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	<u>9</u>	<u>10</u>	1	2	3	<u>4</u>	5	<u>6</u>	7	8	<u>9</u>	<u>10</u>
11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>	11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>
<u>21</u>	<u>22</u>	23	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>	<u>21</u>	<u>22</u>	23	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>
31	<u>32</u>	<u>33</u>	<u>34</u>	<u>35</u>	<u>36</u>	37	<u>38</u>	<u>39</u>	<u>40</u>	31	<u>32</u>	<u>33</u>	<u>34</u>	<u>35</u>	<u>36</u>	37	<u>38</u>	<u>39</u>	<u>40</u>
41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>	41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>
<u>51</u>	<u>52</u>	53	<u>54</u>	<u>55</u>	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>	51	52	53	<u>54</u>	<u>55</u>	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>
61	<u>62</u>	<u>63</u>	<u>64</u>	<u>65</u>	<u>66</u>	67	<u>68</u>	<u>69</u>	<u>70</u>	61	62	<u>63</u>	<u>64</u>	<u>65</u>	<u>66</u>	67	<u>68</u>	<u>69</u>	<u>70</u>
71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>	71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	<u>77</u>	<u>78</u>	79	<u>80</u>
<u>81</u>	<u>82</u>	83	<u>84</u>	<u>85</u>	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>	<u>81</u>	<u>82</u>	83	<u>84</u>	<u>85</u>	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>
91	<u>92</u>	<u>93</u>	<u>94</u>	<u>95</u>	<u>96</u>	97	<u>98</u>	<u>99</u>	<u>100</u>	91	92	93	<u>94</u>	<u>95</u>	<u>96</u>	97	<u>98</u>	<u>99</u>	<u>100</u>

备注: 去掉1, 因为它不是素数

1.3.4 素数的无限性



Euclid
(325 – 265 B.C)

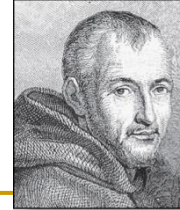
□定理: 存在无限多个素数.

□证明(欧几里得证明法):

- 用反证法. 假设只有有限个 p_1, p_2, \dots, p_n . 令 $q = p_1 p_2 \dots p_n + 1$.
- 根据算术基本定理, q 要么是素数, 要么能被写成两个或多个素数之积. 如果 q 为素数, 那么它又不属于有限个素数中 p_1, p_2, \dots, p_n 矛盾了.
- 如果 q 为合数, 用 q 去除以以上有限个素数中的任何一个素数, 都余1. 也就是说, 根据素数的定义, 以上有限个素数的任何一个, 都不是 q 的因子. 这又和算术基本定理相矛盾. 因此, 存在无限多个.

【备注:有很多证明的方法, 有兴趣的自行查阅更多的文献】

1.3.4 梅森素数



Marin Mersenne
(1588-1648)



华中科技大学
计算机科学与技术学院
School of Computer Science & Technology, HUST

- 人们发现很多素数基本都可以写成 $2^p - 1$ 的形式:
- 【定义】：素数形式为 $2^p - 1$, 其中 p 也是素数, 这种素数叫做**梅森素数**.
 - $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 31$, $2^7 - 1 = 127$ 都是梅森素数.
 - $2^{11} - 1 = 2047$, 不是梅森素数, 因为 $2047 = 23 \cdot 89$.
 - 目前已知最大素数都是梅森素数.
 - 超级计算机可以用来寻找梅森素数.
 - 2013年发现第48个梅森素数, $2^{57885161} - 1$, 该数超过1700万位.
 - 2018年年底发现第51个梅森素数, $2^{82589933} - 1$, 该数有24862048位.
 - 2024年年底发现下一个梅森素数, $2^{136279841} - 1$, 首个使用英伟达GPU找到的梅森素数.
 - 思考: 那为什么要找这些梅森素数呢?

1.3.4 素数的分布

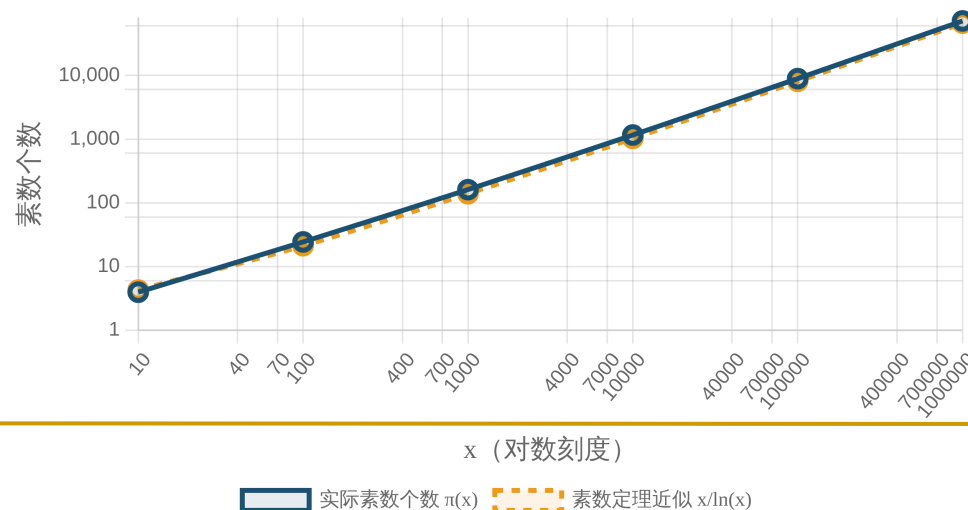
□ 小于一个正整数 x 的素数有多少个呢？素数分布规律是什么呢？

□ **【素数定理】**：当 x 无限增长时，不超过 x 的素数个数与 $x/\ln x$ 之比趋近于1 ($\ln x$ 表示 x 的自然对数, $\log_e x$, $e \approx 2.7182$). 或者另外一种写法：

$$\lim_{n \rightarrow +\infty} \frac{\pi(n)}{n/\ln n} = 1$$

➤ 其中 $\pi(n)$ ：小于等于 n 的素数个数. 定理告诉我们素数的个数可以近似用 $x/\ln x$ 来逼近. 如何证明该定理, 有兴趣的同学自行阅读更多材料.

素数分布与素数定理



1.3.4 产生素数

- 产生一个大素数常用于密码学或其他应用中. 目前, 还没有一个完美的公式 $f(n)$ 使得 $f(n)$ 是素数, 其中 n 表示所有的正整数 n .
- 但是, $f(n) = n^2 - n + 41$ 对于不超过 40 的整数, 都产生对应的素数. 但是 $n=41$ 时, 结果不是素数.
- 幸运地是, 我们可以产生大整数, 其中包含素数(参见后续章节).

1.3.4 关于素数的猜想

- 虽然素数被研究多年, 但还有许多问题一直未解决:
- **哥德巴赫猜想**: 每个大于5的奇数 n 都是三个素数之和(等价于每个大于2的偶数是两个素数之和). 比如 $12=7+5$.
- **孪生素数猜想**: 孪生素数只是相差2的一对素数, 例如3和5, 5和7, 17和19, 4967和4969. 孪生素数猜想断定存在无限多对孪生素数.
- **黎曼猜想**(它与素数分布之间有深刻联系)等

1.3.5 最大公约数

- 【定义】：令 a 和 b 是两个整数, 如果 $d|a$ 和 $d|b$, 则 d 称为 a 和 b 的**公约数**(或称公因子, 公因数).
- 除0以外, 任何整数只有有限个因子. 因而, 两个不全为0的整数只有有限个公约数.
- 【定义】：令 a 和 b 是两个整数, 不全为0. 能使 $d|a$ 和 $d|b$ 的最大整数 d 称为 a 和 b 的**最大公约数**(最大公因数), 记作 $\gcd(a, b)$.
 - \gcd 来自greatest common divisors的首字母缩写.
 - 根据定义可知 $\gcd(a, b)|a$, $\gcd(a, b)|b$

【基础知识：假如整数 \diamond 除以 \diamond 等于一个没有余数的整数(也就是 $\diamond|\diamond$), 那么我们称 \diamond 是 \diamond 的**整数因子**. 比如 $42=6*7$, 因此7是42的因子】

1.3.5 最大公约数

- 因此, 求解两个整数的最大公约数的方法: 可以先找出这两个整数的所有公约数(指能同时整除这两个整数的整数), 然后取其中最大的那个公约数.

- 例:24和36的最大公约数是多少?
- 解:24和36的正公约数是1,2,3,4,6,12, 因此 $\gcd(24,36) = 12$

- 例:17和22的最大公约数是多少?
- 解:17和22除了1以外没有其他正公约数, 因此 $\gcd(17,22) = 1$

1.3.5 最大公约数

- 例:0和22的最大公约数是多少?
- 解:0和22的正公约数是1,2,11,22, 因此 $\gcd(0,22) = 22$
- 备注: $\gcd(0, a) = a$, a 为任意的正整数

- 例:1和22的最大公约数是多少?
- 解:1和22的正公约数有1, 因此 $\gcd(1,22)=1$
- 备注: $\gcd(1, a) = 1$, a 为任意的正整数

1.3.5 最大公约数

- 【定义】：整数 a 和 b 是**互素**的, 如果他们的最大公约数是1.
- 例如之前例子中提到的17和22(因为 $\gcd(17, 22) = 1$)
- 【定义】：整数 a_1, a_2, \dots 是**两两互素**的, 如果当 $1 \leq i < j \leq n$ 时有 $\gcd(a_i, a_j) = 1$.

1.3.5 最大公约数

- 例: 判断10,17,21是否两两互素? 10, 19, 24是否两两互素?
- 解:由于 $\gcd(10,17) = 1$, $\gcd(10,21) = 1$, $\gcd(17,21) = 1$, 所以10, 17, 21是两两互素的; 因为 $\gcd(10,24) = 2$, 所以10, 19, 24不是两两互素的.

1.3.5 最大公约数

□ 求解最大公约数的方法有:

- **方法一(定义法):** 以上求解两个整数的最大公约数, 可以通过定义首先找出两个数的公约数, 然后再找出最大的那个公约数.
- **方法二(因子分解法):** 利用素因子分解式也能求解两个整数的最大公约数.

□ 假设两个正整数 a 和 b 的素因子分解式分别为 $a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$ 其中每个指数都是非负整数, 而且出现在 a 或 b 的素因子分解式中的所有素数都出现在这两个素因子分解式中, 必要时以0指数出现, 那么最大公约数为:

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)}$$

【备注: 证明略, 有兴趣的同学课外自学】

1.3.5 最大公约数

□例: 求120和500的最大公约数.

□解: 因为120和500的素因子分解式为 $120 = 2^3 \cdot 3 \cdot 5$, $500 = 2^2 \cdot 5^3$,
那么 $gcd(120, 500) = 2^{\min(3, 2)} \cdot 3^{\min(1, 0)} \cdot 5^{\min(1, 3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

1.3.5 最小公倍数

- 【定义】： a 和 b 为两个非零的整数, 如果 $a|m$, $b|m$, 则称 m 为 a 与 b 的**公倍数**.
- a 与 b 有无穷多个公倍数.
- 【定义】：正整数 a 和 b 的**最小公倍数**是能被 a 和 b 整除的最小正整数, 记作 $lcm(a, b)$.
 - 备注: lcm 是单词least common multiple的首字母缩写
 - 根据定义可知 $a|lcm(a, b)$, $b|lcm(a, b)$

1.3.5 最小公倍数

□例:24和36的最小公倍数是多少?

□解:24的倍数有24, 48, 72, ... 而36的倍数有36, 72, ... 因此
 $\text{lcm}(24,36)=72$.

□例:1和24的最小公倍数是多少?

□解:1的倍数有1,2,3..., 24的倍数有24,48,72,... 因此 $\text{lcm}(1,24)=24$

□备注: $\text{lcm}(1, a) = a$, a 为任意的正整数

1.3.5 最小公倍数

□ 求解最小公倍数的方法:

- **方法一:** 以上求解两个整数的最小公倍数都是通过定义首先找出两个数各自的倍数, 然后再找出最小的那个公倍数.
- **方法二:** 利用整数的素因子分解式也能求解两个整数的最小公倍数.

□ 假设两个正整数 a 和 b 的素因子分解式分别为:

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}, \quad b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$$

那么最小公倍数为:

$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \cdots p_n^{\max(a_n, b_n)}$$

1.3.5 最小公倍数

□ 例: 求 $2^3 3^5 7^2$ 和 $2^4 3^3$ 的最小公倍数

□ 解: $\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} 3^{\max(5,3)} 7^{\max(2,0)} = 2^4 3^5 7^2$

1.3.5 最小公倍数

- 两个正整数 a 和 b 的最大公约数和最小公倍数之间的关系如下:
- 定理:令 a 和 b 为正整数, 则 $ab = \gcd(a, b) \cdot \text{lcm}(a, b)$
- 备注:该定理的证明略.

1.3.5 最小公倍数

□例: 证明若 $a|m$, $b|m$, 则 $\text{lcm}(a,b)|m$.

□证:

- 记 $M = \text{lcm}(a, b)$.
- 先假设 M 不能整除 m , 那么设 $m = qM + r$, $0 < r < M$.
- 由 $a|m$, $a|M$ (备注:根据最小公倍数的定义, $a|\text{lcm}(a,b)$), 及 $r = m - qM$, 可推出 $a|r$ (根据推论:如果 a, b, c 是整数, 其中 $a \neq 0$, 使得 $a|b$ 和 $a|c$, 那么当 m 和 n 是整数时, 有 $a|mb + nc$).
- 同理, 有 $b|r$.
- 因为 $a|r$, 同时 $b|r$, 那么 r 是 a 和 b 的公倍数.
- 但是我们已知 $r < M$, 这和刚才假设的 M 是 a 和 b 的最小公倍数先矛盾.
- 因此 $\text{lcm}(a, b)$ 能整除 m , 得证.

1.3.6 欧几里得算法

□ 求解最大公约数的方法有:

- 方法一(定义法): 以上求解两个整数的最大公约数, 可以通过定义首先找出两个数的公约数, 然后再找出最大的那个公约数.
- 方法二(因子分解法): 利用素因子分解式也能求解两个整数的最大公约数.
- **方法三(欧几里得算法):** 直接从整数的素因子分解式来计算两个正整数的最大公约数的效率很低, 因为寻找素因子分解式就很耗时. 其实, 欧几里得算法给出了一种更高效的做法.

□ 在讲欧几里得(Euclid)算法之前, 首先我们先看一个简单的例子:

1.3.6 欧几里得算法

□例: 求 $\gcd(91, 287)$

□解:

- $287 = 91 \cdot 3 + 14$. 如果存在任意的一个公约数 x , $x|287$, $x|91$, 那么一定会存在 $x|(287-91 \cdot 3)=x|14$ (来自推论: a, b, c 是整数, 且 a 不为0, 当 $a|b$, $a|c$, 那么当 m 和 n 为整数时, 有 $a|mb + nc$).
- 如果存在任意的一个公约数 y , $y|91$, $y|14$, 那么一定会存在 $y|287 = y|(91 \cdot 3 + 14)$.
- 因此, 287和91的最大公约数, 91和14的最大公约数, 两个最大公约数将是同一个数.
- 那么求287和91的最大公约数变为求解91和14的最大公约数.
- 类似地, $91 = 14 \cdot 6 + 7$, 而 $14 = 7 \cdot 2 + 0$
- 因此 $\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = \gcd(7, 0) = 7$