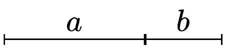
有趣的贵金属比 Ratios

两个数字的比率中，有一个被认为是最美丽的。最著名的就是**黄金比 1.618**，在艺术设计，建筑设计，自然界形成的花卉等中都有很好的体现。

除了黄金分割比外，还有很多根据贵金属名称来命名的比，如 白银比，青铜比，镍比等，统称为 **贵金属比。**

1、Golden Ratio 黄金比、黄金分割

将一条线段分成 $a, b\ (a > b)$ 两段，且满足 长边：短边=全长：长边，即 $a : b = (a+b) : a,\ a^2 = b(a+b)$, 分割如下图：



令 $r = \dfrac{a}{b}$, 则有：

$r = 1 + \dfrac{b}{a} = 1 + \dfrac{1}{r},\ r^2 = r + 1$

解得正数解为 $r = \dfrac{1+\sqrt{5}}{2} \approx 1.618033988749895...,\ \dfrac{1}{r} = \dfrac{\sqrt{5}-1}{2} \approx 0.6180339887498949...$

上述 r 值就是黄金分割数，记作 $\varphi = \dfrac{1+\sqrt{5}}{2}$

2、Silver Ratio 白银比

将一条线段分成 $a, a, b\ (a > b)$ 三段，且满足 长边：短边 = 全长：长边，即 $a : b = (2a+b) : a,\ a^2 = b(2a+b)$, 分割如下图：

令 $r = \dfrac{a}{b}$, 则有：

$r = 2 + \dfrac{b}{a} = 2 + \dfrac{1}{r},\ r^2 = 2r + 1$

解得正数解为 $r = 1 + \sqrt{2} \approx 2.414213562373095...$

上述 r 值就是白银分割数，记作 $\sigma = 1 + \sqrt{2}$

3、Mentallic Ratios 贵金属比

将一条线段分成 $a, \cdots, a, b\ (a > b),\ n+1$ 段，且满足 长边：短边 = 全长：长边，即 $a : b = (na+b) : a,\ a^2 = b(na+b)$, 分割如下图：

令 $\lambda_n = \dfrac{a}{b}$, 则有：

$\lambda_n = n + \dfrac{b}{a} = n + \dfrac{1}{\lambda_n},\ \lambda_n^2 = n\lambda_n + 1$

解得正数解为 $\lambda_n = \dfrac{n + \sqrt{n^2+4}}{2}$

上述 λ_n 值就是金属分割数。显然 $\lambda_1 = \varphi, \lambda_2 = \sigma$

也有定义：

- $n = 3, \lambda_3$ 为 Bronze Ratio 青铜分割；
- $n = 4, \lambda_4$ 为 Copper Ratio 铜分割；
- $n = 5, \lambda_5$ 为 Nickel Ratio 镍分割；

注意：以上三个重要比都是通过几何方法得到的，采用几何中的线段之比来定义相关分割比。直观简单，作图简洁，易于理解。

4、列表总结

$Name$	n	$Equation$	$Value$	$Continued\ fraction$	$OEISlink$
$Platinum$	0	$(0+\sqrt{4})/2$	1	$-$	$-$
$Golden$	1	$(1+\sqrt{5})/2$	$1.618033988749895...$	$[1;1,1,1,1,1,1,1,1,1,1...]$	$OEIS:A001622$
$Silver$	2	$(2+\sqrt{8})/2$	$2.414213562373095...$	$[2;2,2,2,2,2,2,2,2,2,2...]$	$OEIS:A014176$
$Bronze$	3	$(3+\sqrt{13})/2$	$3.302775637731995...$	$[3;3,3,3,3,3,3,3,3,3,3...]$	$OEIS:A098316$
$Copper$	4	$(4+\sqrt{20})/2$	$4.23606797749979...$	$[4;4,4,4,4,4,4,4,4,4,4...]$	$OEIS:A098317$
$Nickel$	5	$(5+\sqrt{29})/2$	$5.192582403567252...$	$[5;5,5,5,5,5,5,5,5,5,5...]$	$OEIS:A098318$
$Aluminum$	6	$(6+\sqrt{40})/2$	$6.16227766016838...$	$[6;6,6,6,6,6,6,6,6,6,6...]$	$OEIS:A176398$
$Iron$	7	$(7+\sqrt{53})/2$	$7.140054944640259...$	$[7;7,7,7,7,7,7,7,7,7,7...]$	$OEIS:A176439$
Tin	8	$(8+\sqrt{68})/2$	$8.123105625617661...$	$[8;8,8,8,8,8,8,8,8,8,8...]$	$OEIS:A176458$
$Lead$	9	$(9+\sqrt{85})/2$	$9.109772228646444...$	$[9;9,9,9,9,9,9,9,9,9,9...]$	$OEIS:A176522$

5、与Fibonacci sequence（斐波那契数列）的关系

传统Lucas Sequence序列形如下：
$x_n = P \times x_{n-1} - Q \times x_{n-2}$

令 $P = b, Q = -1$, 就得到 Mentallic Ratios：
$x_n = b \times x_{n-1} + x_{n-2}$

为了避免除数为零，令 头两个数为 $x_1 = 1$， $x_2 = 1$.
当 $b = 1$, 我们得到斐波那契数列：
$x_n = x_{n-1} + x_{n-2}, \mapsto \{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \cdots\}$

```
[sympy.fibonacci(x) for x in range(15)]   #获取前15个斐波那契数列
```

当 $b = 2$, 我们得到数列：
$x_n = 2 \times x_{n-1} + x_{n-2}, \mapsto \{1, 1, 3, 7, 17, 41, 99, 239, 577, 1393, \cdots\}$

类似数列还有 `bell, bernoulli, catalan, euler, harmonic, lucas, genocchi, partition, tribonacci`

6、Python Source Code

```
from itertools import count, islice
from _pydecimal import getcontext, Decimal

def metallic_ratio(b):
```

```
    m, n = 1, 1
    while True:
        yield m, n
        m, n = m*b + n, m

def stable(b, prec):
    def to_decimal(b):
        for m,n in metallic_ratio(b):
            yield Decimal(m)/Decimal(n)

    getcontext().prec = prec
    last = 0
    for i,x in zip(count(), to_decimal(b)):
        if x == last:
            print(f'after {i} iterations:\n\t{x}')
            break
        last = x

for b in range(4):
    coefs = [n for _,n in islice(metallic_ratio(b), 15)]
    print(f'\nb = {b}: {coefs}')
    stable(b, 32)

print(f'\nb = 1 with 256 digits:')
stable(1, 256)
```

7、圆族 A Family of Circles

圆族的圆心序列为 $(0,b_n)$, $b_1=r_1,b_n=\sqrt{2}b_1$;
圆族的半径序列为 $r_n=\sqrt{2}r_{n-1},r_1$ 给定。

在GeoGebra中，可以通过迭代列表命令产生点列：

`Plist = Iterationlist((sqrt(2) x(p), y(p)-x(p)),p,{(r_1,0)},n)`

其中 `n = Slider(1,10,1), r_1 = Slider(0.5,5)`

再根据上述点列画圆族：

`Circlist = Zip(Circle((0, y(p)), x(p)), p, Plist)`

相关半径画法：

`Radlist = Zip(Segment((0, y(p)), p), p, Plist)`

相关半径画法：

`Radlist = Zip(Segment((0, y(p)), p), p, Plist)`

`Slopelist = Zip(Segment((0, y(p)-x(p)), p), p, Plist)`