对于小数点后确定只需保留几位的场景，

比如超市 xxx.xx 元，用定点数表示

法中的 BCD 码即可，定点意味着小

数点位置固定，比如始终把十进制

数最后两位视做小数部分，点在

倒数第三位。

BCD 码用 4 位二进制表示一个 10 进

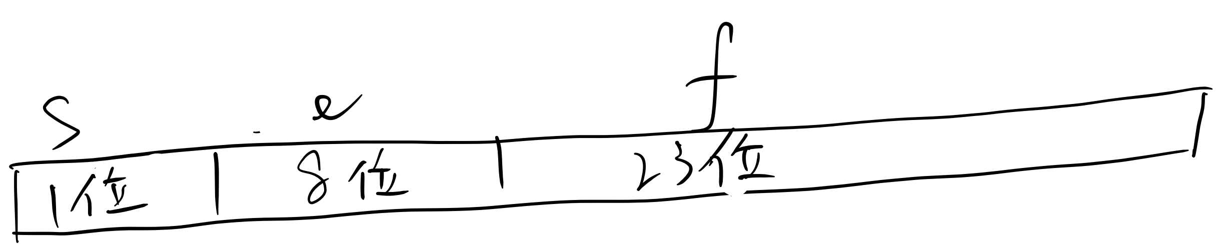
制数，在准确表达的同时，也牺牲了

可用范围（16 个数只用 10 个），且在

用场景限制较大。

对于小数点在保留位数不确定，或者表示的数范围很大时，定点数法就不适用了。

浮点数的小数点位置不确定，由科学计数法中10的指数来控制，以下实例为证：（32位浮点数）



非规数表示：

非规格数：

$number = (-1)^s \times 2^e \times 1.f$

当 $e=0$ 且 $f=0$ 时 number 没定 $\rightarrow 0$

$e=0$ 且 $f \neq 0$ 时 $number \xrightarrow{没定} 0.f$

$e=255$ 时范指无穷大或负无穷大(看s)

S必用一位，剩下31位中，e用得越多，可表示范围越大，有效位越少；反之e越小，可表示范围越小，有效位越多。综合考虑，IEEE标准规定为e用8位，f用23位.

$2^{23} = 8388608$，十进制 7位有效数

2^{126} ($2^8 \Rightarrow -127 \sim 126$) $= 4 \times 10^{37}$，约为表示范围

十进制浮点数 $\longleftrightarrow$ 二进制

① 2 → 10

6 5 4 3 2 1 0 -1 -2 -3 -4 -5

1001010.10001

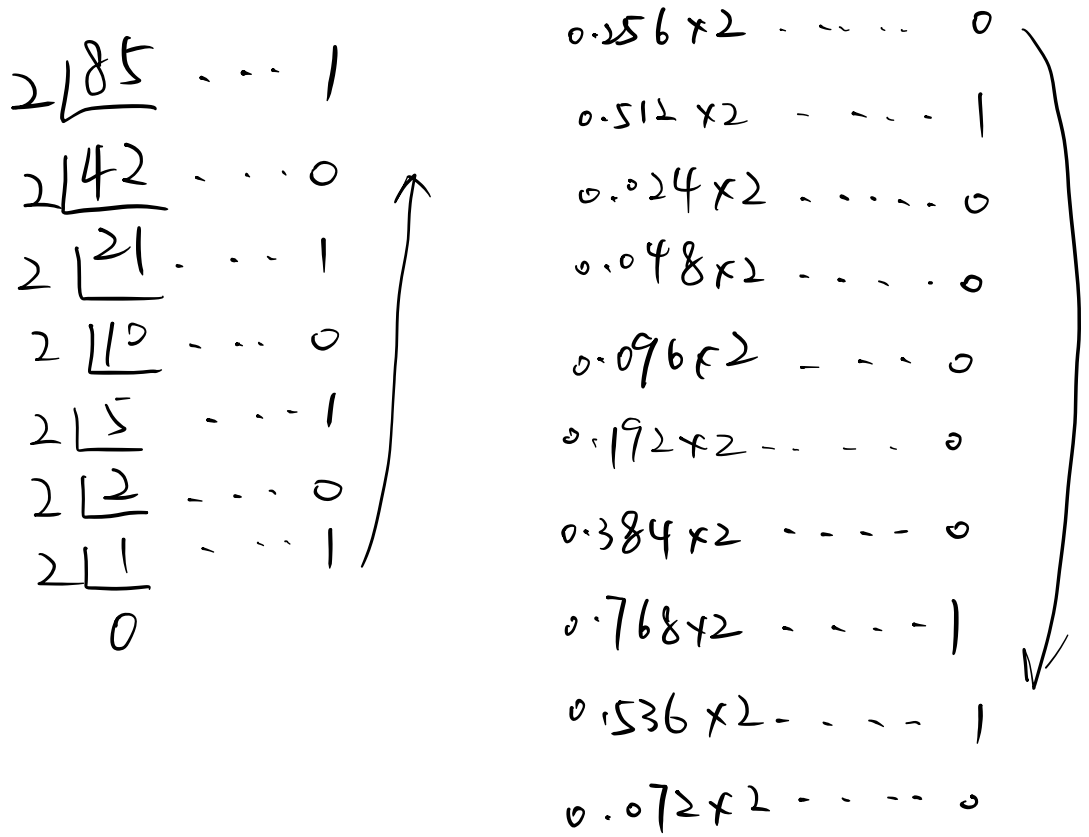
$= 2^6 + 2^3 + 2^1 + 2^{-1} + 2^{-5}$

$= 64 + 8 + 2 + 0.5 + 0.03125$

$= 74.53125$

② 10 → 2

$85.256 = 1010101.01000011\ldots$



精度损失.

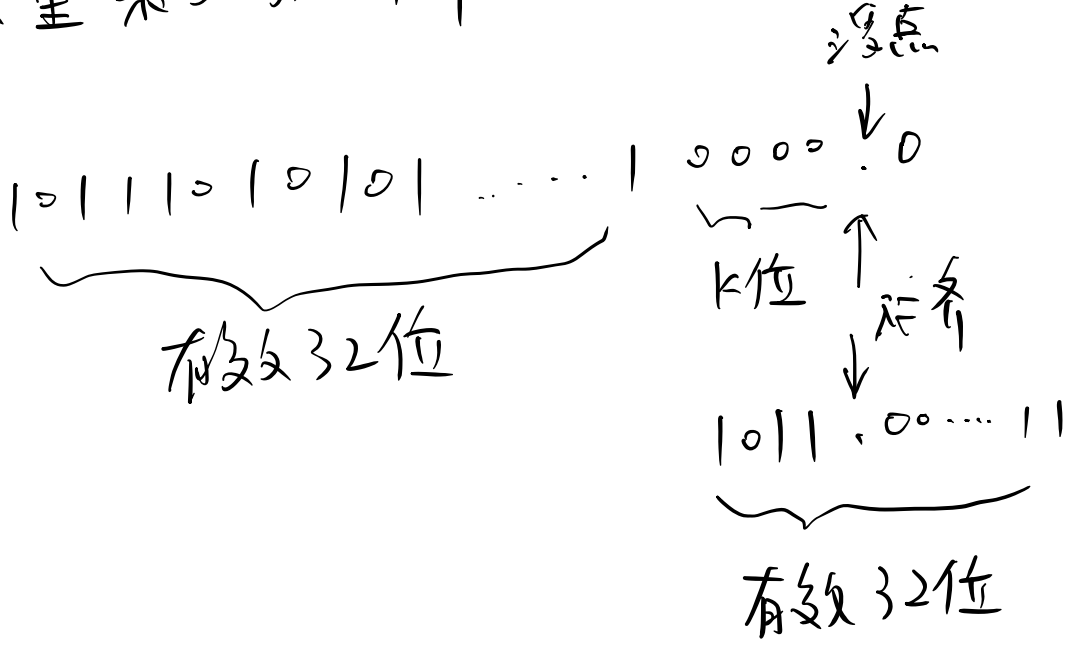
浮点有效位23位时，进行加法时要

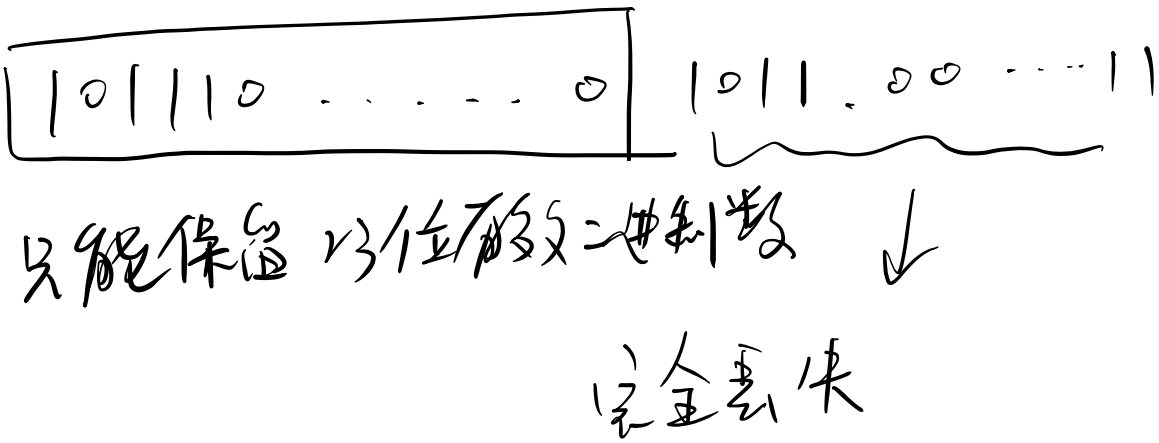
.业占对齐 若两数量级相差太大，对

小数点对齐

舍弃小的一个精度会部分甚至完全丢失，

在大量累加统计中间就很大.





解决方案：

两个e判断小者会损失的精度，提前将损失部份摘出，放在另一个内存地址中，又加上不会损失部份，每次都将摘出部份加在一起，最后统计时再将摘出和加回总和中。