万能欧几里得算法

算法简介

一种 $O(\log n)$ 计算形如 $f(n)=\sum_{i=1}^n \lfloor \frac {ai+b}c \rfloor$ 的函数的算法。

算法原理

不妨考虑待计算函数就是 $f(n)=\sum_{i=1}^n \lfloor \frac {ai+b}c\rfloor$□同时假设 $b\lt c$□考虑按如下过程计算 $f(n)$□

考虑在 $(0,n]$ 范围内逐渐增加 x□当 $y=\frac {ax+b}c$ 恰好等于某个整数时 $y\gets y+1$□称为 U 操作。

当 x 恰好等于某个整数时有 $\text{ans}\gets \text{ans}+y$□称为 R 操作。如果遇到整点，先进行 U 操作再进行 R 操作。

于是计算 $f(n)$ 等价于处理由 $y=\frac {ax+b}c(x\in (0,n])$ 产生的操作序列 S□

先考虑如何得到这个操作序列，设 $g(a,b,c,n,U,R)$ 表示由 $y=\frac {ax+b}c(x\in (0,n])$ 产生的操作序列 S□

定义字符串幂操作 $S^k=S^{k-1}S$□

如果 $a\ge c$□显然 R 操作前面至少有 $\lfloor \frac ac\rfloor$ 个 U 操作。将这 $\lfloor \frac ac\rfloor$ 个 U 操作和一个 R 操作视为一个整体，于是有

$$ g(a,b,c,n,U,R)=g(a\bmod c,b,c,n,U,U^{\lfloor \frac ac\rfloor}R) $$

如果 $b\ge c$□显然有

$$ g(a,b,c,n,U,R)=U^{\lfloor \frac bc\rfloor}g(a,b\bmod c,c,n,U,R) $$

最后，如果 $c\gt \max (a,b)$□则考虑构造 $y=\frac {ax+b}c$ 按 $y=x$ 做对称变换得到的直线 $y=\frac {cx-b}a$□

不难发现□$y=\frac {cx-b}a$ 产生的序列和 $y=\frac{ax+b}c$ 产生的序列大致是互补的，即 U/R 互补。

但要处理遇到整点的情况，此时两条直线都是先 U 再 R 不满足互补关系，考虑将 $y=\frac {cx-b}a$ 整体向下偏移得到 $y=\frac {cx-b-1}a$□

这样原先在 $y=\frac {cx-b}a$ 先 U 再 R 的整点等价于 $y=\frac {cx-b-1}a$ 先 R 在 U□

但偏移也导致 $y=\frac {cx-b-1}a$ 的边界发生了变化，考虑暴力处理边界，设 $m=\frac {an+b}c$□于是有

$$ g(a,b,c,n,U,R)=R^{\lfloor \cfrac{c-b-1}{a}\rfloor}Ug(c,(c-b-1)\bmod a,a,m-1,R,U)R^{n-\lfloor \cfrac{cm-b-1}{a}\rfloor} $$

假如不考虑字符串拼接等操作的复杂度，上述式子可以 $O(\log n)$ 计算。

接下来考虑如何通过得到的字符串即操作序列来计算答案。事实上可以在拼接字符串时维护答案。

设 $h(S)$ 表示字符串 S 对应的答案，两个串分别为 $S1,S2$□$dx=\text{cntR}(S1),dy=\text{cntU}(S1),n=\text{cntR}(S2)$□

$$ h(S1S2)=h(S1)+\sum_{i=1}^n (\lfloor \frac {ai+b}c\rfloor+dy)=h(S1)+\sum_{i=1}^n (\lfloor \frac {ai+b}c\rfloor)+n\times dy=h(S1)+h(S2)+n\times dy $$