数学篇

1. 杜教筛 $O(n^{\frac{2}{3}})$

○前置技能:

1° 积性函数: $f(p\cdot q) = f(p)\cdot f(q)$. $\gcd(p,q)=1$.

若 $\forall p,q$, $f(p,q)=f(p)\cdot f(q)$ → 完全积性函数.

常见积性函数: $\mu(n)$ — 莫比乌斯函数　　$d(n)$ — 约数个数

$\varphi(n)$ — 欧拉函数　　$\sigma(n)$ — 约数和函数

完全积性函数: $\epsilon(n)$ — 一元函数　$\epsilon(n)=[n=1]$

$I(n)$ — 恒等函数　$I(n)=1$.

$id(n)$ — 单位函数　$id(n)=n$.

2° 狄利克雷卷积.

$$(f*g)(n) = \sum_{d|n} f(d)*g(\tfrac{n}{d})$$

$$f*\epsilon = f$$

3° 整除分块

对于式子 $\sum_{d=1}^{n} f(\lfloor \frac{n}{d} \rfloor)$

① $\lfloor \frac{N}{d} \rfloor$ 最多 $2\sqrt{N}$ 种取值　　② $\lfloor \frac{N}{i} \rfloor$ 与 $\lfloor \frac{N}{i} \rfloor$ 相等, 则 $i_{max} = \left\lfloor \frac{N}{\lfloor \frac{N}{i} \rfloor} \right\rfloor$

∴
```
for (int i=1; i<=N ){
    j = N/(N/i);
    ans += f(N/i) × (j-i+1);
    i=j+1;
}
```

○杜教筛：计算 $\sum_{i=1}^{n} f(i)$ 　　$f(i)$ 为积性函数.

构造两个积性函数 h、g.

$h=f*g$ 　　$S(n)=\sum_{i=1}^{n} f(i)$

$$\sum_{i=1}^{n} h(i)=\sum_{i=1}^{n}\sum_{d\mid i} g(d)\cdot f\left(\frac{i}{d}\right)$$

$\vdots$

$$g(1)\cdot S(n)=\underbrace{\sum_{i=1}^{n} h(i)}_{\text{较短时间求出}}-\underbrace{\sum_{d=2}^{n} g(d)\cdot S\left(\left\lfloor\frac{n}{d}\right\rfloor\right)}_{\text{整除分块}}$$

○应用

1° $S(n)=\sum_{i=1}^{n} i*\varphi(i)$.

$\therefore h(n)=\sum_{d\mid n} d*\varphi(d)*g\left(\frac{n}{d}\right)$

$\therefore$ 令 $g\left(\frac{n}{d}\right)=\frac{n}{d}$

$\therefore h(n)=\sum_{d\mid n} n\cdot\varphi(d)=n^2$

$\therefore\ S(n)=\sum_{i=1}^{n} i^2-\sum_{d=2}^{n} d\cdot S\left(\left\lfloor\frac{n}{d}\right\rfloor\right)$ 　　∴成功.

∴预处理 $n^{\frac{2}{3}}$ 个 $S(n)$，后面的整除分块做，

以及用 unordered_map 存下来后面算过的答案.

2、莫比乌斯反演

内容：若 $f(n)=\sum_{d|n} g(d)$

则 $g(n)=\sum_{d|n}\mu(d) f(\frac{n}{d})$

两个结论： $n=\sum_{i|n}\varphi(i)$

$[n=1]=\sum_{i|n}\mu(i)$

例：求 $\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)==1]$

解：原式 $=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|gcd(i,j)}\mu(d)$

$=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\substack{d|i\\ d|j}}\mu(d)$

$=\sum_{d}\mu(d)\times\lfloor\frac{n}{d}\rfloor\times\lfloor\frac{m}{d}\rfloor$

3、广义欧拉降幂

$$a^b \equiv \begin{cases} a^{b\%\phi(p)} & \gcd(a,p)=1 \\ a^b & \gcd(a,p)\neq 1,\ b<\phi(p) \\ a^{b\%\phi(p)+\phi(p)} & \gcd(a,p)\neq 1,\ b\geq\phi(p) \end{cases} \quad (\%p)$$

4、

// Pollard-Rho 大数因式分解

// 蔡勒公式：　//判周五.　　$c=y/100$　$y=y\%100$

若小于等于 1582.10.4：

$$w = y + [\tfrac{y}{4}] + [\tfrac{c}{4}] - 2c + [\tfrac{13(m+1)}{5}] + d + 2 \quad (\%7)$$

大于 1582.10.4

$$w = [\tfrac{c}{4}] - 2c + y + [\tfrac{y}{4}] + [\tfrac{13(m+1)}{5}] + d - 1 \quad (\%7)$$

// 利用 bitset.　　//若干个 0、1 的数位，可用于压位.

bitset：bitset<120>　开一个 120 位的 bitset.

bitset<120> f.

f.reset();　所有数位置为 0.

f.set(i);　第 i 位赋为 1.

f.reset(i);　第 i 位赋为 0.

类欧几里德专题.

① $\frac{a}{b} < \frac{x}{y} < \frac{c}{d}$

求 $\min(x,y)$.

求 $\min(y,x)$. ⟶ 把式子先倒一下即可.

1° $a=0$: $\begin{cases} x=1 \\ y=\lceil \frac{d}{c} \rceil \end{cases}$

2° $a<b$: $\frac{d}{c} < \frac{y}{x} < \frac{b}{a}$

3° $\frac{a\%b}{b} < \frac{x-\lfloor\frac{a}{b}\rfloor y}{y} < \frac{c-\lfloor\frac{a}{b}\rfloor d}{d}$

solve!

② 求 $\sum_{i=1}^{n} \lfloor \frac{c+bi}{a} \rfloor$

$\sum c/a + c\%a + \frac{b}{a}i + [(b\%a)i]\%a.$

$= \sum(\frac{c}{a}+\frac{b}{a}i) + [c\%a+(b\%a)i]\%a.$

$\sum(\frac{c}{a}+\frac{b}{a}i) + \sum \lfloor \frac{c+b\%a\,i}{a} \rfloor \% a.$

杂项

1、python:

①
```python
while True:
    try:
    except:
        break.
```

②
```python
L =[[0 for i in range(m+1)] for j in range(n+1)]
```

③输出不换行: print(x, end="") print('%d ... %d...' % (x1, x2))

④ L= list (map (int, input().split())).

⑤ sort: L.sort(key=lambda x:(x[0]:x[1]))

⑥全局变量: global x

2、Time:

```cpp
clock_t start, ends;
start = clock();
...
ends = clock();
cout<<(double)(ends-start)/CLOCKS_PER_SEC<<endl;
```

开始建造后缀自动机

准备工作：建立数组ch, fa, len，准备指针$last, cnt$。SAM的构造方法是不断地向已经建好的SAM中加入新的节点。$last$表示上一个被插入的节点，cnt表示SAM中的节点数量。一开始，$last = cnt = 1$，表示只有一个起点的初始SAM。

接下来，假设要往SAM里加入一个字符x。

1. 新建节点$np = ++cnt$。新建节点p。$p = last$。$last = np$。
2. 如果不存在$ch[p][x]$，令$ch[p][x] = np, p = fa[p]$。重复此步骤。
3. 如果到最后还没有一个p拥有儿子x，令$fa[np] = 1$。退出过程。
4. 当$ch[p][x]$出现时，令$q = ch[p][x]$。如果$len[q] == len[p] + 1$，令$fa[np] = q$。退出过程。
5. 否则有点麻烦。新建节点$nq = ++cnt$，将q的儿子都复制给nq，令$len[nq] = len[p] + 1$。
6. 令$fa[nq] = fa[q], fa[q] = fa[np] = nq$。
7. 从p开始沿着后缀链接，将所有$ch[p][x] == q$的节点的$ch[p][x]$都替换成nq。

将你的字符串的所有字符都一一进行如上操作后，你就得到了用你的字符串构建出来的SAM。

你不需要知道为什么这么操作可以得到SAM，你只需要记下以下的代码，做几道题强化记忆，然后就可以用SAM的性质来秒题了。

```
void insert(int x)
{
    int np=++cnt,p=last;
    len[np]=len[p]+1,last=np;
    while(p&&!ch[p][x])ch[p][x]=np,p=fa[p];
    if(!p)fa[np]=1;
    else
    {
        int q=ch[p][x];
        if(len[q]==len[p]+1)fa[np]=q;
        else
        {
            int nq=++cnt;len[nq]=len[p]+1;
            memmove(ch[nq],ch[q],sizeof(ch[nq]));
            fa[nq]=fa[q],fa[np]=fa[q]=nq;
            while(ch[p][x]==q)ch[p][x]=nq,p=fa[p];
        }
    }
}
```

后缀自动机的奇妙性质

现在，你已经拥有SAM了，你需要知道它有什么用。这里列举了SAM的一些基本且常用的性质。

请牢记以下每一条内容！都十分有用！不要去问"为什么是这样的"！（如果一定要问，请参照上文蓝色放大的Warning）

首先，SAM的点数与边数都是$O(n)$的。*记住，由于每次插入最多新建两个点，所以应该开字符总量两倍的空间。*计算空间时别忘了你开了26倍的ch数组。

在SAM上从起点开始沿着边随便走走，得到的一定是子串。同时，每一个子串都可以在SAM上走出一条唯一对应的路径。也就是说，子串和SAM上从起点开始的每一条路径一一对应。路径数等于子串数。

起点可以看做是代表空串的点。

重点：定义子串的$right$集合：这个子串在原串中所有出现的位置的右端点的集合。

比如说：AAAABBAAAAABAAABBAA

子串AAB出现了3次，右端点集合为$\{5, 12, 16\}$。这就是子串AAB的$right$集合。

一个节点能够代表的所有子串的$right$集合是一样的。$right$集合相等的子串一定被同一个节点代表。（所以，我们会使用"节点的$right$集合"这个说法。）两个节点的$right$集合之间要么真包含，要么没有交集。若节点y的$right$集合包含了节点x的$right$集合，那么y能代表的子串均为x能代表的子串的真后缀。

重点：定义节点x的后缀链接$fa(x)$：如果有一些节点的$right$集合包含了x的$right$集合，$fa(x)$是其中$right$集合的大小最小的那一个。

后缀链接们组成了一棵"后缀链接树"（不是后缀树）。后缀链接树的根为起点。若节点y的$right$集合包含了节点x的$right$集合，那么y在后缀链接树上是x的祖先。

一个节点的$right$集合等于他在后缀链接树上的所有儿子的$right$集合的并集。而且儿子的$right$集合之间两两没有交集。

每个节点能代表的子串的长度范围是一段连续的区间。这很好理解，因为它们的结束位置都是相同的。

我们求出每个节点能代表的最长串的长度（即$len(x)$）了，那最短长度呢？其实就等于后缀父亲节点的$len + 1$。也就是说，所有本质不同的子串的数量等于$\sum len(x) - len(fa(x))$。

总结

以上就是SAM的基本性质~对于一道特定的题，你可能需要通过上面的性质推出你需要的新性质。如果你还有什么疑问可以向我留言，我（在退役前）会在一天之内回复的！（你也可以去问更强的boshi和litble，别去问zyf因为他已经退役了。）

题单我就不给了，因为网上有很多很多。。。