Descartes' theorem（笛卡尔定理）

若平面上四个半径为 r_1、r_2、r_3、r_4 的圆两两相切于不同点，则其半径满足以下结论：

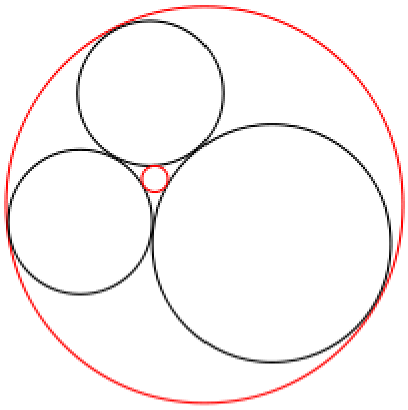
（1）若四圆两两外切，则

$$\left(\sum_{i=1}^{4}\frac{1}{r_i}\right)^2=2\sum_{i=1}^{4}\frac{1}{r_i^2}$$

；

（2）若半径为 r_1、r_2、r_3 的圆内切于半径为 r_4 的圆中，则

$$\left(\frac{1}{r_1}+\frac{1}{r_2}+\frac{1}{r_3}-\frac{1}{r_4}\right)^2=2\sum_{i=1}^{4}\frac{1}{r_i^2}$$



皮克定理

皮克定理是指一个计算点阵中顶点在格点上的多边形面积公式，该公式可以表示为 2S=2a+b-2，其中 a 表示多边形内部的点数，b 表示多边形边界上的点数，s 表示多边形的面积。

图论

网络流模型

1、最大权闭合子图

构图：正权点连 S，负权点连 T，图上的所有边（单向，双向）连 inf。

图上的边=>约束关系，即选 A 必选 B，不选 A 也不能选 B。

答案：所有正权和-最小割

子图：在残量网络中从 S 出发，一遍 dfs，走还有容量的边，经过的点就是要选的点。

2、分数规划

例：最大密度子图（最大化|E|/|V|）

构图：二分答案 k，所有边连 S，边权 1，所有点连 T，边权-k，跑最大权闭合子图。

(u->v)=>([uv]->u),([uv]->v)

最大化 A/B，二分答案 k，A-B*k>0，构图跑最大权闭合子图。

3、最小点路径覆盖（最少的路径数覆盖所有点）

构图：把点 i 拆成 i 和 i'，S->i 连 1，i'->T 连 1。对于图上的每一条边，a->b'连 1。

答案=点数-最大匹配

4、最小边路径覆盖（最少的路径数覆盖所有边）

答案：有向无环图中最小边路径覆盖的值等于图中所有“入少出多”的点的（入度-出度）之和。

5、最小顶点覆盖（用最少的点，让每条边都至少和其中一个点关联）

最小顶点覆盖=最大匹配

6、二分图最大独立集（选一些顶点，这些顶点间两两没有连线）

构图：最大独立集=顶点个数-最小顶点覆盖（最大匹配）

7、原图最大独立集=补图最大团

上下界网络流

1、无源无汇上下界最大流

上界 r，下界 l。新建源汇 ST。

du[i]=in[i]（i 节点所有入流下界之和）-out[i]（i 节点所有出流下界之和）

x->y 连 r-l。

if du[i]>0 S->i 连 du[i] else i->T 连-du[i]。

2、有源有汇上下界最大流

上界 r，下界 l。源 S 汇 T，超级源 SS 超级汇 TT

du[i]=in[i]（i 节点所有入流下界之和）-out[i]（i 节点所有出流下界之和）

x->y 连 r-l。

if du[i]>0 SS->i 连 du[i] else i->TT 连-du[i]。

T->S 连 inf,判可行流，得最大流 ans1。

后去掉 SS,TT，去掉 T->S,在残余网络上跑最大流 ans2。

ans=ans1+ans2。

3、有源有汇上下界最小流

按照可行流建图，然后先不连 T 到 S 流量为的 inf 边。

跑一遍最大流。

再加上那条边，再跑一遍最大流。

后一次跑的答案就是最小流。

差分约束构图

Xa-Xb<=c   Xb->Xa 连边权 c 跑最短路 负环无解。
Xa-Xb<c → Xa-Xb<=c-1
Xa==Xb → Xa-Xb<=0 && Xa-Xb>=0

如果题目要求对于部分 Xi 确定 Xi 的值为 Ai，那么建立 0 节点，X0=0。
对于要求定值的，Xi-0=Ai，0->Xi 连 Ai，Xi->0 连-Ai。对于不确定的，0->Xi 连 inf。

2-SAT

选择 xi 点代表 Ai=1(选择此物品)，选择 yi 点代表 Ai=0(不选择此物品)
Ai AND Aj=0：两条边 xi→yj,xj→yi
Ai OR Aj=1：两条边 yi→xj,yj→xi
Ai OR(NOT Aj)=1：两条边 xj→xi,yi→yj
Ai XOR Aj=1：四条边 xi→yj,xj→yi,yi→xj,yj→xi
Ai XOR Aj=0：四条边 xi→xj,xj→xi,yi→yj,yj→yi
Ai=1：把 yi 删掉，所以一条边 yi→xi
Ai=0：把 xi 删掉，所以一条边 xi→yi

Hall 定理：

若一个二分图是 k-正则二分图，则该图存在 k 个不相交的完备匹配
K-正则二分图：每个点的度均为 k

关于欧拉回路

无向图存在欧拉回路的充要条件

一个无向图存在欧拉回路，当且仅当该图所有顶点度数都为偶数,且该图是连通图。

有向图存在欧拉回路的充要条件

一个有向图存在欧拉回路，所有顶点的入度等于出度且该图是连通图。

混合图存在欧拉回路条件

要判断一个混合图 G（V,E）（既有有向边又有无向边）是欧拉图，方法如下：
假设有一张图有向图 G'，在不论方向的情况下它与 G 同构。并且 G'包含了 G 的所有有向边。那么如果存在一个图 G'使得 G'存在欧拉回路，那么 G 就存在欧拉回路。
其思路就将混合图转换成有向图判断。实现的时候，我们使用网络流的模型。现任意构造一个 G'。用 Ii 表示第 i 个点的入度，Oi 表示第 i 个点的出度。如果存在一个点 k，|Ok-Ik|mod 2=1，那么 G 不存在欧拉回路。接下来则对于所有 Ii>Oi 的点从源点连到 i 一条容量为(Ii-Oi)/2 的边，对于所有 Ii<Oi 的点从 i 连到汇点一条容量为(Oi-Ii)/2 的边。如果对于节点 U 和 V，无向边（U，V）∈E，那么 U 和 V 之间互相建立容量为 1 的边。如果此网络的最大流等于∑|Ii-Oi|/2，那么就存在欧拉回路。

数论

中国剩余定理

用现代数学的语言来说明的话，中国剩余定理给出了以下的一元线性同余方程组：

$$(S):\quad \begin{cases} x \equiv a_1 & \pmod{m_1} \\ x \equiv a_2 & \pmod{m_2} \\ \quad\vdots \\ x \equiv a_n & \pmod{m_n} \end{cases}$$

有解的判定条件，并用构造法给出了在有解情况下解的具体形式。

中国剩余定理说明：假设整数$m_1,m_2,\dots,m_n$两两互质，则对任意的整数：$a_1,a_2,\dots,a_n$，方程组(S)有解，并且通解可以用如下方式构造得到：

设 $M = m_1 \times m_2 \times \cdots \times m_n = \prod_{i=1}^{n} m_i$ 是整数$m_1,m_2,\dots,m_n$的乘积，并设 $M_i = M/m_i,\ \forall i \in \{1,2,\cdots,n\}$ 是除了m_i以外的$n-1$个整数的乘积。

设 $t_i = M_i^{-1}$ 为 M_i 模 m_i 的数论倒数（t_i 为 M_i 模 m_i 意义下的逆元）$M_i t_i \equiv 1 \pmod{m_i},\ \forall i \in \{1,2,\cdots,n\}.$

方程组(S)的通解形式为 $x = a_1 t_1 M_1 + a_2 t_2 M_2 + \cdots + a_n t_n M_n + kM = kM + \sum_{i=1}^{n} a_i t_i M_i, \quad k \in \mathbb{Z}.$

在模 M 的意义下，方程组(S)只有一个解：$x = \left(\sum_{i=1}^{n} a_i t_i M_i\right) mod M$

原根

a 位模 p 意义下的原根
a^i%p!=a^j%p(i!=j)(0<i,j<=p-1) --> a^(p-1)==1(mod p) 当且仅当 i==p-1

暴力求原根：从 a=2 开始枚举，暴力判断 a^(p-1)==1(mod p)是否当且仅当 i==p-1

不暴力的求原根：求出 p-1 所有不同的质因子 p1,p2…pm，对于任何 2<=a<=p-1,判定 a 是否为 p 的原根，只需要检验 a^((p-1)/p1),a^((p-1)/p2),…a^((p-1)/pm)这 n 个数中，是否存在一个数 mod p 为 1，若存在，a 不是 x 的原根，否则就是 x 的原根。

p 有原根充要条件：p=1,2,4,p,2p,p^n(p==prime(except 2))

x*y%p ==> g^i==x(mod p),g^j==y(mod p) ==> x*y%p --> (i+j)%(p-1)
其中 g 为 mod p 意义下的原根

关于质因数分解

可以在线性筛的时候处理出每个数字最大的质因子，把质因数分解复杂度从 sqrt 级降到 log 级

有关 gcd 的数论总结

（一）1≤x,y≤n，求$gcd(x,y)=1$ 的(x,y)对数。

$$ans = 2*(\sum_{i=2}^{n}\varphi(i))+1$$

（二）1≤x≤n，1≤y≤m，求$gcd(x,y)=1$ 的(x,y)对数。

$$ans = \sum_{i=1}^{\min(n,m)} \mu(i)\left[\frac{n}{i}\right]\left[\frac{m}{i}\right]$$

```
for(int l=1,r;l<=n;l=r+1){
    r=min(n/(n/l),m/(m/l)); //总共有 sqrt(n)段 l，r 区间
    ans[pos]+=(n/l)*(m/l)*(sum[r]-sum[l-1]); //sum 为 u[i]前缀和
}
```

（三）1≤x≤n，1≤y≤m，求$gcd(x,y)=d$的(x,y)对数。

问题可以转化为：$1 \leqslant x \leqslant \left[\frac{n}{d}\right]$，$1 \leqslant y \leqslant \left[\frac{m}{d}\right]$，求$gcd(x,y)=1$的(x,y)对数。

（四）1≤x≤n，1≤y≤m，求$\sum_x\sum_y gcd(x,y)$。

首先，我们得先知道一个结论：$\sum_{d|n}\varphi(d)=n$

$$\sum_x\sum_y gcd(x,y) = \sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{d|\,\gcd(x,y))}\varphi(d)$$

$$=\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{d|\mathrm{x}\ \&\ \mathrm{d}|\mathrm{y})}\varphi(d)$$

$$=\sum_{d=1}^{\min(n,m)}\varphi(d)\left[\frac{n}{d}\right]\left[\frac{m}{d}\right]$$

后同（二）

（五）1≤x≤n，1≤y≤m，求 gcd(x,y)=p（p 为质数）的(x,y)对数。

枚举所有的 p，结合(三)可得

$$ans = \sum_{p}\sum_{i=1}^{\min\left(\frac{n}{\mathrm{p}},\frac{m}{p}\right)}\mu(i)\left[\frac{n}{pi}\right]\left[\frac{m}{pi}\right]$$

//左式涉及一个结论：$\left[\frac{\left[\frac{n}{x}\right]}{y}\right]=\left[\frac{n}{xy}\right]$

令$T = p * i$，考虑对于每个$\left\lfloor \frac{n}{T} \right\rfloor \left\lfloor \frac{m}{T} \right\rfloor$对答案的贡献是多少，也就是要求它的系数是多少。那么，对于一个 T，每当枚举到它的某个质因子 p 时，总会有$i = \frac{T}{p}$，使答案加上$\mu\left(\frac{T}{p}\right)\left\lfloor \frac{n}{T} \right\rfloor \left\lfloor \frac{m}{T} \right\rfloor$。因此，如果枚举所有的 T，那么就有

$$ans = \sum_{T=2}^{\min(n,m)} \left(\left\lfloor \frac{n}{T} \right\rfloor \left\lfloor \frac{m}{T} \right\rfloor \sum_{p} \mu\left(\frac{T}{p}\right) \right)$$

令$g(T) = \sum_p \mu\left(\frac{T}{p}\right)$，如果能够预处理出所有的$g(i)$，并求出其前缀和$s(i)$，那么我们也能每次$O(\sqrt{n})$求答案了。

分析一下$g(T)$的性质，由$\mu(i)$的定义，易得到

$$g(T) = \begin{cases} (-1)^{k-1} * k, & T = p_1 p_2 p_3 \ldots\ldots p_k \\ (-1)^{k}, & T = p_1^2 p_2 p_3 \ldots\ldots p_k \\ 0 & \text{其他所有情况} \end{cases}$$

```
void Euler()
{
      for(int i=2;i<=N;i++)
      {
          if(!check[i]) prime[++tot]=i,s[i]=1;
          for(int j=1;j<=tot;j++)
          {
              int k=i*prime[j]; if(k>N) break;
              check[k]=true;
              if(i%prime[j])//不整除
              {
                  if(check[i] && abs(s[i])==1) s[k]=-s[i];
                     else s[k]=s[i]>0?-s[i]-1:abs(s[i])+1;
                  if(!s[i]) s[k]=0;
              }
              else {s[k]=(check[i] && abs(s[i])<=1)?0:-s[i]/abs(s[i]); break;}
          }
      }
      for(int i=1;i<=N;i++) s[i]=s[i-1]+s[i];
}
```

其中，数组 s[i]前部分表示 g[i]，后部分表示 g[i]的前缀和。

一些定理：

(1) $\sum_{d|n}\varphi(d)=n$

(2) $\left[\frac{\left[\frac{n}{x}\right]}{y}\right]=\left[\frac{\frac{n}{x}}{y}\right]=\left[\frac{n}{xy}\right]$

(3) $\sum_{i\&1=1}C_n^i=\sum_{i\&1=0}C_n^i$

n 个元素选取 k 个元素，k 为奇数的方案数与 k 为偶数的方案数相同。

(4) $\sum_{gcd(x,n)=1}x=\frac{n\varphi(n)}{2}$

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

(6) 莫比乌斯反演: $g(n)=\sum_{d|n}f(d)\Leftrightarrow f(n)=\sum_{d|n}\mu(d)g\left(\frac{n}{d}\right)$

计数类问题

卡特兰数

通项公式一 $C_n=\frac{1}{n+1}C_{2n}^{n}=C_{2n}^{n}-C_{2n}^{n-1}$；

通项公式二 $C_n=\frac{1}{n+1}\sum_{i=0}^{n}(C_n^i)^2$；

递推公式一 $C_{n+1}=\frac{2(2n+1)}{n+2}C_n$，且 $C_0=1$；

递推公式二 $C_{n+1}=\sum_{i=0}^{n}C_iC_{n-i}$，且 $C_0=1$；

打表：1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670, 129644790, 477638700, 1767263190, 6564120420, 24466267020, 91482563640, 343059613650, 1289904147324, 4861946401452, ...

例题：n 个数字 1-n 按顺序进栈，求出栈序列种类数

二阶：n*m，从（1,1）到（n,m）=C（n+m,n）

要求在 y=x 直线下：c(n+m,n)-2*c(n+m-1,n)

第一类 string 数

$$s(i,j)=s(i-1,j)*(i-1)+s(i-1,j-1)$$

$$x*(x+1)*(x+2)*\ldots*(x+n-1)=\sum_{i=1}^{n}s(n,i)*x\hat{\ }i$$

第二类 string 数

$$s(i,j)=s(i-1,j)*j+s(i-1,j-1)=\frac{\sum_{k=0}^{j}(-1)^k*C(j,k)*(j-k)^i}{j!}$$

定义：s(n,m)表示把 n 个元素划分成 m 个无序集合的方案数

如果要求有序集合的话后面乘上 $j!$ 就好了。

根据化成卷积得式子可以 O(nlogn)求出 s(i,1),s(i,2)···s(i,i)

贝尔（bell）数

定义：把 n 个元素划分成若干个无序集合的方案数

打表：1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975

$$f[i]=\sum_{j=0}^{i}s(i,j)=\sum_{k=0}^{i-1}C(i-1,k)*f[k]$$

Burnside 定理

枚举每一种置换，在置换下不相同的方案数

$$\mathrm{L} = \frac{\sum_{i=1}^{n} d[i]}{n}$$

其中 n 表示置换的个数，d[i]表示置换 i 下不变的方案数

Polya 定理

枚举每一种置换，在置换下不相同的方案数

$$\mathrm{L} = \frac{\sum_{i=1}^{n} m^{c[i]}}{n}$$

其中 n 表示置换的个数，d[i]表示置换 i 下不变的方案数，m 表示颜色种类数，c[i]表示置换 i 的循环个数

Matrix-tree 定理（无向图生成树计数）

构建度数矩阵 A[i][j]，其中 A[i][i]表示 i 点的度数，A[i][j] =0(i!=j)
构建邻接矩阵 B[i][j]，B[i][j]=1(I,j 有连边)
构建基尔霍夫矩阵 C=A-B，求 C 的 n-1 阶子式行列式的值的绝对值（把最后一行最后一列去掉）就是答案
//有重边的时候也可以做，B[i][j]等于边数
//A[i][i]=ΣA[i][k](k!=i)