**1：无源汇上下界可行流（循环流）：**

对于每根管子有一个上界容量up和一个下界容量low，我们让这根管子的容量下界变为0，上界为up-low。

可是这样做了的话流量就不守恒了，为了再次满足流量守恒，即每个节点"入流=出流”，我们增设一个超级源点st和一个超级终点sd。我们开设一个数组du[]来记录每个节点的流量情况。

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）。

当du[i]大于0的时候，st到i连一条流量为du[i]的边。

当du[i]小于0的时候，i到sd连一条流量为-du[i]的边。

最后对（st，sd）求一次最大流即可，当所有附加边全部满流时（即maxflow==所有du[]>0之和），有**可行解**。

**2：有源汇上下界可行流：**

模型:现在的网络有一个**源点s和汇点t**,求出一个流使得源点的总流出量等于汇点的总流入量,其他的点满足流量守恒,而且每条边的流量满足上界和下界限制.

源点s的流出量等于汇点t的流入量,我们就可以从汇点t向源点s连一条下界为0上界为无穷大的边,相当于把从源点s流出的流量再流回来.在这样的图中套用上面的算法求出一个可行的循环流,拆掉从汇点t到源点s的边就得到一个可行的有源汇流.

**3：有源汇上下界最大流：**

模型:现在的网络有一个源点s和汇点t,求出一个流使得源点的总流出量等于汇点的总流入量,其他的点满足流量守恒,而且每条边的流量满足上界和下界限制.在这些前提下要求**总流量最大**.

套用上面的算法求出一个有源汇有上下界可行流.此时的流不一定最大.

接下来在残量网络上跑s-t最大流即可.(删除与超源超汇相连的边)

最终的最大流流量=可行流流量(即t到s的无穷边上跑出的流量)( head[t]^1 )+新增广出的s-t流量

**4：有源汇上下界最小流：**

模型:现在的网络有一个源点s和汇点t,求出一个流使得源点的总流出量等于汇点的总流入量,其他的点满足流量守恒,而且每条边的流量满足上界和下界限制.在这些前提下要求**总流量最小**.

理解一下dinic的反向边.反向边的流量增加等价于正向边的的流量减少.因此我们在残量网络上找出t到s的流就相当于减小了s到t的流,因此我们在跑出可行流的残量网络上跑t-s最大流,用可行流的大小减去这一次t-s最大流的大小就是最小流的大小.(t-s最大流其实是尽量缩减s-t方向的流).

ans+=dinic.max\_flow();

dinic.addedge(t,s,inf,0);

ans+=dinic.max\_flow();

5：最大费用循环流

　　用最小费用流求最大费用循环流时，解决负环的一种方法：

（1）先将所有边权取反。

（2）建边。正权值的边容量为1，费用为权值。负权值的边u->v拆成3条边，分别是S->v，v->u，u->T，容量都为1，v->u费用为负权的相反数，其他2条费用为0。这样会出现某个点有多条边连到S或T，可以互相抵消到一方为0为止，统计剩下多少条k，将其中1条的容量设为k，其他的全部删掉。如果全部抵消掉了，那就将连S和T的边全部删掉。（这个删边的方法有技巧）

（3）跑一次最小费用流得到的总费用，加上所有负权之和之后（注：此时答案已为负的），再取反即得到最大费用。