上下界网络流建图

可行流：

无源点和汇点：建立一个超级源点SS，和一个超级汇点TT，对于一条边（U，V，(L~R)）来说，我们将其拆成三条边：

1）（SS，V，L）

2)（U，TT，L）

3）（U，V，R-L）

有源点和汇点：（S,T为原图的源点和汇点）其他所有边跟无源点和汇点的情况一样连边。然后为了保证流量平衡，再加三条边：

1）（SS，S，0）

2)（T，TT，0）

3）（T，S，INF）

如果所有从SS流出的流量都满流，则这种状态可行

如果求最大流只需要在残量网络跑S到T的最大流再加上所有边的下界就行了。

Java大数开根号:

static BigInteger SQRT(BigInteger x){

        BigInteger l = BigInteger.ONE ;

        BigInteger r = x ;

        BigInteger temp = BigInteger.ZERO ;

        while(!l.equals(r)){

            BigInteger mid=(l.add(r)).divide(BigInteger.valueOf(2)) ;

            if(temp.compareTo(BigInteger.ZERO)!=0&&temp.compareTo(mid)==0){

                break ;

            }else{

                temp = mid ;

            }

            if(temp.compareTo(BigInteger.ZERO)==0){

                temp = mid ;

            }

            if(mid.multiply(mid).compareTo(x)==1){

                r=mid ;

            }else{

                l=mid ;

            }

        }

        if(l.multiply(l).compareTo(x)==1){

            l=l.subtract(BigInteger.ONE) ;

        }

        return l;

    }

Java格式：

import java.util.\*;

import java.math.\*;

public class Main{

static BigInteger INF=new BigInteger("100000000000000000000000000");

static Scanner cin=new Scanner (System.in);

static BigInteger[] x=new BigInteger[2004];

public static void main(String[] args) {

int T;

while(cin.hasNext())w33ha();

}

}