POJ 2239 Selecting Courses Solution

write by Gestalti Lur
2012-08-08

题目大意

给出 N 种课程的上课时间（每种课只要上其中一节就可以），问最多能上多少节课

算法分析

以课程-上课时间构图然后计算最大匹配即可。上课时间对应的序号可以 hash 统计或者用不重复的公式计算。

参考代码

JAVA

```
/*
  POJ 2239
  test KM
  2012-07-18
  2012-07-20
  ACCEPTED
 */
import java.io.*;
import java.util.*;

public class poj2239{

    static final int MAXN = 402;
    static final int INF = ( 1 << 30 ) - 1;
    static int n , m;
    static int lx[] = new int[ MAXN ];
    static int ly[] = new int[ MAXN ];
    static int linky[] = new int[ MAXN ];
    static boolean visx[] = new boolean[ MAXN ];
    static boolean visy[] = new boolean[ MAXN ];
    static int w[][] = new int[ MAXN ][ MAXN ];
    static int slack[] = new int[ MAXN ];

    static int h[][][] = new int [ 8 ][ 13 ][ 402 ];
    static int ccnt[][] = new int [ 8 ][ 13 ];

    static Scanner in = new Scanner(System.in);
    static PrintStream out = System.out;
```

```
public static void init(){
  int i , j , tp , tq , req;

  n = in.nextInt();;

  for( i = 0 ; i < MAXN ; ++ i ){
      Arrays.fill( w[ i ] , 0 );
  }
  for( i = 1 ; i <= 7 ; ++ i )
      for( j = 1 ; j <= 12 ; ++ j )
        ccnt[ i ][ j ] = 0;
  Arrays.fill( lx , 0 );
  Arrays.fill( ly , 0 );
  Arrays.fill( linky , -1 );

  for( i = 1 ; i <= n ; ++ i ){
      req = in.nextInt();
      for( j = 1 ; j <= req ; ++ j ){
        tp = in.nextInt();
        tq = in.nextInt();
        h[ tp ][ tq ][ ++ ccnt[ tp ][ tq ] ] = i;
      }
  }

  m = 0;
  for( i = 1 ; i <= 7 ; ++ i )
      for( j = 1 ; j <= 12 ; ++ j )
        if( ccnt[ i ][ j ] > 0 ){
            ++ m;
            for( tp = 1 ; tp <= ccnt[ i ][ j ] ; ++ tp ){
              w[ h[ i ][ j ][ tp ] ][ m ] = 1;
            }
        }
  n = n > m ? n : m;
}

public static void out(){
  int i , cnt = 0;
  /*
  for( i = 1 ; i <= n ; ++ i )
      out.printf( "%d " , linky[ i ] );
  out.println();
  */
  for( i = 1 ; i <= n; ++ i )
      if( linky[ i ] != -1 )
        cnt += w[ linky[ i ] ][ i ];
  out.printf( "%d\n" , cnt );
}
```

```
public static boolean find( int x ){
  visx[ x ] = true;
  for( int y = 1 ; y <= n ; ++ y ){
      if( visy[ y ] == true )
        continue;
      int t = lx[ x ] + ly[ y ] - w[ x ][ y ];

      if( t == 0 ){
        visy[ y ] = true;
        if( linky[ y ] == -1 || find( linky[ y ] ) ){
            linky[ y ] = x;
            return true;
        }
      }
      else if( slack[ y ] > t )
        slack[ y ] = t;
  }
  return false;
}

public static void KM(){
  for( int i = 1 ; i <= n ; ++ i )
      for( int j = 1 ; j <= n ; ++ j )
        lx[ i ] = w[ i ][ j ] > lx[ i ] ? w[ i ][ j ] : lx[ i ];
  for( int x = 1 ; x <= n ; ++ x ){
      for( int i = 1 ; i <= n ; ++ i )
        slack[ i ] = INF;
      while( true ){
        Arrays.fill( visx , false );
        Arrays.fill( visy , false );
        if( find( x ) )
            break;
        int d = INF;
        for( int i = 1 ; i <= n ; ++ i )
            if( visy[ i ] == false && d > slack[ i ] )
              d = slack[ i ];
        for( int i = 1 ; i <= n ; ++ i ){
            if( visx[ i ] == true )
              lx[ i ] -= d;
            if( visy[ i ] == true )
              ly[ i ] += d;
            else
              slack[ i ] -= d;
        }
      }

  }
```

```
   }

   public static void main( String args[] )
     throws Exception {
     while( in.hasNext() ){
         init();
         //out.println( "init ok" );
         KM();
         //out.println( "KM OK" );
         out();
     }
   }
}
/*
3
2 2 2 3 3
1 2 2
1 3 3
6
4 1 1 2 2 3 3 4 4
2 1 1 2 2
1 1 1
1 2 2
1 3 3
2 1 1 4 4
*/
```

Pascal

```
{ACCEPTED}
PROGRAM POJ2239;
CONST
  MAXN   = 300;
  INF  = MAXINT;
VAR
  N , M            : INTEGER;
  W                : ARRAY[ 1..MAXN , 1..MAXN ] OF INTEGER;
  LX , LY , LINKY , SLACK : ARRAY[ 1..MAXN ] OF INTEGER;
  VISX , VISY        : ARRAY[ 1..MAXN ] OF BOOLEAN;
  H            : ARRAY[ 1..7 , 1..12 , 1..MAXN ] OF INTEGER;
  CCNT               : ARRAY[ 1..7 , 1..12 ] OF INTEGER;

PROCEDURE INIT();
VAR
  I , J , REQ , P , Q : INTEGER;
BEGIN
  FILLCHAR( CCNT , SIZEOF( CCNT ) , 0 );
  FILLCHAR( W , SIZEOF( W ) , 0 );
  FILLCHAR( LX , SIZEOF( LX ) , 0 );
  FILLCHAR( LY , SIZEOF( LY ) , 0 );
  FILLCHAR( LINKY , SIZEOF( LINKY ) , 0 );

  READLN( N );
  FOR I := 1 TO N DO BEGIN
    READ( REQ );
    FOR J := 1 TO REQ DO BEGIN
     READ( P , Q );
     INC( CCNT[ P , Q ] );
     H[ P , Q , CCNT[ P , Q ] ] := I;
    END;
    READLN;
  END;

  M := 0;
  FOR P := 1 TO 7 DO BEGIN
    FOR Q := 1 TO 12 DO BEGIN
     IF CCNT[ P , Q ] <> 0 THEN BEGIN
       INC( M );
       FOR I := 1 TO CCNT[ P , Q ] DO
         W[ H[ P , Q , I ] , M ] := 1;
     END;
    END;
  END;

  IF M > N THEN
    N := M;
  {FOR P := 1 TO N DO BEGIN
    FOR Q := 1 TO N DO
     WRITE( W[ P , Q ] , ' ' );
    WRITELN;
  END;}

END; { INIT }

FUNCTION FIND(X : INTEGER ):BOOLEAN;
VAR
  Y , T : INTEGER;
BEGIN
  VISX[ X ] := TRUE;
  FOR Y := 1 TO N DO BEGIN
    IF VISY[ Y ] THEN
     CONTINUE;
    T := LX[ X ] + LY[ Y ] - W[ X , Y ];
    IF T = 0 THEN BEGIN
     VISY[ Y ] := TRUE;
     IF ( LINKY[ Y ] = 0 ) OR ( FIND( LINKY[ Y ] ) ) THEN BEGIN
```

```
        LINKY[ Y ] := X;
        EXIT( TRUE );
      END;
     END
     ELSE IF SLACK[ Y ] > T THEN
      SLACK[ Y ] := T;
  END;
  EXIT( FALSE );
END; { FIND }

PROCEDURE KM();
VAR
  I , J , D , X : INTEGER;
BEGIN
  FOR I := 1 TO N DO
    FOR J := 1 TO N DO
     IF LX[ I ] < W[ I , J ] THEN
       LX[ I ] := W[ I , J ];

  FOR X := 1 TO N DO BEGIN
    FOR I := 1 TO N DO
     SLACK[ I ] := INF;
    WHILE ( TRUE ) DO BEGIN
     FILLCHAR( VISX , SIZEOF( VISX ) , FALSE );
     FILLCHAR( VISY , SIZEOF( VISY ) , FALSE );
     IF FIND( X ) THEN
       BREAK;
     D := INF;
     FOR I := 1 TO N DO
       IF D > SLACK[ I ] THEN
         D := SLACK[ I ];
     FOR I := 1 TO N DO BEGIN
       IF VISX[ I ] THEN
         DEC( LX[ I ] , D );
       IF VISY[ I ] THEN
         INC( LY[ I ] , D )
       ELSE
         DEC( SLACK[ I ] , D );
     END;
    END;
  END;
END; { KM }

PROCEDURE OUT();
VAR
  I , CNT : INTEGER;
BEGIN
  {FOR I := 1 TO N DO
```

```
    WRITE( LINKY[ I ] , ' ' );
  WRITELN;}
  CNT := 0;
  FOR I := 1 TO N DO
    IF LINKY[ I ] <> 0 THEN
     INC( CNT , W[ LINKY[ I ] , I ] );
  WRITELN( CNT );
END;

BEGIN
  WHILE NOT EOF DO BEGIN
    INIT;
    KM;
    OUT;
  END;
END.
```