

2010 Fuzhou Onsite B "Nubulsa Expo" Solution

本作品採用[知識共享署名-非商业性使用-相同方式共享 3.0 Unported 許可協議](#)进行許可
write by Gestalti Lur

2012-09-21

題目鏈接: <http://acm.fzu.edu.cn/problem.php?pid=2001>
SPOJ FZ10B

題目大意

給出一個 $N(2 \leq N \leq 300)$ 個點和 $M(M \leq 50000)$ 條邊的有向帶權圖求其最小割。

算法分析

參考 Stoer-Wagner¹ 介紹的最小割算法。

參考代碼

PS: SPOJ 上數據可能需要使用 long long 或者 __int 64.

```
*
2010 FUZHOU REGIONAL ONSITE B
Stoer-Wagner Algorithm test
2012-09-20
ACCEPTED
gestapolur
*/
#include<cstdio>
#include<cstring>
#define MAXN 313
#define INF 2141483647
int n , m , s , cnt;
int w[ MAXN ][ MAXN ] , mark[ MAXN ];
bool in[ MAXN ];

bool init()
{
    scanf("%d%d%d" , &n , &m , &s );
    if( n == 0 && m == 0 && s == 0 )
        return false;
    int i , j , u , v , c;
    for( i = 1 ; i <= n ; ++ i )
        for( j = 1 ; j <= n ; ++ j )
            w[ i ][ j ] = 0;
    for( i = 1 ; i <= m ; ++ i )
    {
        scanf("%d%d%d" , &u , &v , &c );
```

¹ Journal of the ACM, Vol. 44, No. 4, July 1997, pp. 585-591.

```

    if( u != v )
        w[ u ][ v ] += c , w[ v ][ u ] += c;
    }
    return true;
}

void mincut()
{
    int i , j , k , l1 , l2 , minc = INF , tminc , tmax , tot = n;
    for( i = 1 ; i <= n ; ++ i )
    {
        for( tminc = 0 , j = 1 ; j <= n ; ++ j )
            tminc += w[ i ][ j ];
        minc = tminc < minc ? tminc : minc;
    }
    if( n > 2 )//mincutphase
        for( l1 = 0 , l2 = 1 , i = 1 ; i < tot ; ++ i , -- n )
        {
            memset( in , false , sizeof( bool ) * ( n + 1 ) );
            memcpy( mark , w[ 1 ] , sizeof( int ) * ( n + 1 ) );
            for( in[ 1 ] = true , mark[ 1 ] = 0 , j = 1 ; j < n ; ++ j )
            {
                for( l1 = l2 , l2 = 0 , tmax = 0 , k = 1 ; k <= n ; ++ k )
                    if( !in[ k ] && tmax < mark[ k ] )
                        { tmax = mark[ k ]; l2 = k; }
                if( !l2 ) { printf( "0\n" ); return ; }//graph were not connected
                in[ l2 ] = true;
                for( k = 1 ; k <= n ; ++ k )
                    if( mark[ k ] < w[ l2 ][ k ] )
                        mark[ k ] = w[ l2 ][ k ];
            }
            for( j = 1 ; j <= n ; ++ j )//merge
            {
                if( j != l1 ) { w[ l1 ][ j ] += w[ l2 ][ j ]; w[ j ][ l1 ] += w[ j ][ l2 ]; }
                w[ l2 ][ j ] = w[ n ][ j ]; w[ j ][ l2 ] = w[ j ][ n ];
            }
            for( tminc = 0 , j = 1 ; j < n ; ++ j )//count the number
                tminc += w[ l1 ][ j ];
            minc = tminc < minc ? tminc : minc;
        }
    printf( "%d\n" , m < tot - 1 ? 0 : minc );
    return ;
}

int main()
{
    while( init() )
        mincut();
}

```

```
return 0;  
}
```