

差分约束系统

Cpt.

定义

如果一个系统由 n 个变量和 m 个约束条件组成，形成 m 个形如 $a_i - a_j \leq k$ 的不等式($i, j \in [1, n]$, k 为常数)，则称其为差分约束系统。亦即，差分约束系统是求解关于一组变量的特殊不等式组的方法。

求解差分约束系统，可以转化成图论的单源最短路径（或最长路径）问题。

通俗一点地说，差分约束系统就是一些不等式的组，而我们的目标是通过给定的约束不等式组求出最大值或者最小值或者差分约束系统是否有解。

特征

差分约束系统有且只有两个元素和一个实数构成。

$$(x-y \geq k)$$

差分约束系统的应用很广，都会有一定的背景，我们只需要根据题意构造出差分约束系统，然后再根据题目的要求求解就行了。

一般题目会有三种情况：(1)、求取最短路 (2)、求取最长路 (3)、判断差分约束系统的解是否存在当然这三种也可能会相互结合。

差分约束的求解

将题目中给出的不等式关系转化为有向图中的边。

如：

$x_1 > x_2 + d$ $x_2 \rightarrow x_1$ ，边权为 $d+1$

$x_1 < x_2 + d$ $x_1 \rightarrow x_2$ ，边权为 $-d-1$

$x_1 \geq x_2 + d$ $x_2 \rightarrow x_1$ ，边权为 d

$x_1 \leq x_2 + d$ $x_1 \rightarrow x_2$ ，边权为 $-d$

$x_1 = x_2$ $x_1 \leftrightarrow x_2$ 边权为 0

转化为有向图后跑图论算法就可以求解。

但要用SPFA之类的算法判负环。（无解的情况）

BZOJ2330

Description

幼儿园里有 N 个小朋友， lx 老师现在想要给这些小朋友们分配糖果，要求每个小朋友都要分到糖果。但是小朋友们也有嫉妒心，总是会提出一些要求，比如小明不希望小红分到的糖果比他的多，于是在分配糖果的时候， lx 需要满足小朋友们的 K 个要求。幼儿园的糖果总是有限的， lx 想知道他至少需要准备多少个糖果，才能使得每个小朋友都能够分到糖果，并且满足小朋友们所有的要求。

BZOJ2330

Input

输入的第一行是两个整数 N , K 。

接下来 K 行, 表示这些点需要满足的关系, 每行3个数字, X , A , B 。

如果 $X=1$, 表示第 A 个小朋友分到的糖果必须和第 B 个小朋友分到的糖果一样多;

如果 $X=2$, 表示第 A 个小朋友分到的糖果必须少于第 B 个小朋友分到的糖果;

如果 $X=3$, 表示第 A 个小朋友分到的糖果必须不少于第 B 个小朋友分到的糖果;

如果 $X=4$, 表示第 A 个小朋友分到的糖果必须多于第 B 个小朋友分到的糖果;

如果 $X=5$, 表示第 A 个小朋友分到的糖果必须不多于第 B 个小朋友分到的糖果;

Output

输出一行, 表示 $lxhgww$ 老师至少需要准备的糖果数, 如果不能满足小朋友们的所有要求, 就输出 -1 。

Solution

看题目可知这是一道差分约束的题目。

根据每种关系建边如下： $X=1 A=B A \leftrightarrow B:0$

$X=2 A < B A \rightarrow B:1$

$X=3 A \geq B A \leftarrow B:0$

$X=4 A > B A \leftarrow B:1$

$X=5 A \leq B A \rightarrow B:0$

对于每种情况建边，然后跑一边SPFA。（最长路）

因为可能会有自环或环的情况，都不可能存在。

跑SPFA时记录入队次数，超过N弹出。

SPFA的dist起始值为1， $ans = \sum dist[i]$

对于每个点做一遍SPFA会超时，所以将所有点放入队列中，所有点一起跑SPFA。

Solution

```
for(i=1;i<=K;i++){
    int o=read(),x=read(),y=read();
    switch(o){
        case 1:P.add(x,y,0),P.add(y,x,0);break;
        case 2:P.add(x,y,1);break;
        case 3:P.add(y,x,0);break;
        case 4:P.add(y,x,1);break;
        case 5:P.add(x,y,0);break;
    }
}
```


Solution

```
void SPFA()
{
    int h=0,t=0;
    memset(into,0,sizeof into);
    for(int i=1;i<=N;i++)l[++t]=i,into[i]++; //入队++
    while(h<t){
        int front=l[++h];
        flag[front]=0;
        for(int i=P.head[front];i!=-1;i=P.nxt[i]){
            if(dist[P.To[i]]<P.W[i]+dist[front]){//求最长路
                dist[P.To[i]]=P.W[i]+dist[front];
                if(!flag[P.To[i]]){
                    l[++t]=P.To[i],flag[P.To[i]]=1;
                    into[P.To[i]]++;
                    if(into[P.To[i]]>N){SF=-1;return ;} //判环
                }
            }
        }
    }
    return ;
}
```

Tips

类似的题目：[BZOJ3436](#)

GL&HF