作业九-PB18030980

Question1 2-SAT

问题描述

现在给出n个bool变量$x_1, x_2, x_3 ... x_n$,对于任意一个bool类型的变量，要不取true，要不取false

现在给出m个约束条件$f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), .. f_m(x_1, x_2, \ldots, x_n)$

这个问题要我们回答的是是否存在一个bool变量的取值集合，使得这个集合满足所有m个约束条件

问题抽象

现在我们将n个变量拓展为$2 \times n$个变量$, x_1, !x_1, x_2, !x_2..!x_n, x_n$,显然，当$x_i = true$时，$!x_i = false, elsewhen !x_i = true, x_i = false$

我们将这2n个bool量抽象为一个图上的$2 \times n$个顶点，设为$y_1, y_2, y_3, .. y_{2\times n-1}, y_{2\times n}$，假设$x_i --> y_{2\times i-1}, !x_i --> y_{2\times i}$

将约束条件抽象为图上的有向边，若$y_i = true$则$y_j = true$抽象为一个从顶点y_i指向顶点y_j的有向边，简单地说，若存在y_i到y_j的有向边，那么y_j的真假性必须与y_i一致

显然，我们可以得到这个结论，现在这个有向图被分成若干个强连通分量，$G_1, G_2, .. G_n$在任何一个强连通分量中，这个分量中的顶点必须真假性完全相同,若任何一个i，我们都可以得到x_i与$!x_i$不在同一个强连通分量中，则这种取值存在，反之不存在

问题变成了找有向图的强连通分量的问题

算法概述

算法分为以下几步

- 使用**Tarjan**算法计算出有向图中的强连通分量，时间复杂度是O(N+M),N是顶点数，M是边数
- 对于所有x_i和$!x_i$判断，他们是否在同一个强连通分量中

最终时间复杂度是O(N+M)，M是与m有关的多项式

Question 2

中国科学技术大学

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA

Hefei,Anhui.230026 The People's Republic of China

先构造一个找 S_a 集合的算法

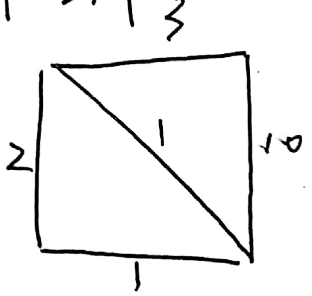
将边按照边权排序 $w_1 < w_2 < \cdots < w_n$

~~令~~ 令 i=n 首先 w_n 属于 S_a，将 w_n ~~的~~ 关联的两个顶点 i, j 打上标记，即 $w_n = ij$ tag(i)=true tag(j)=true => step 0（初始化）

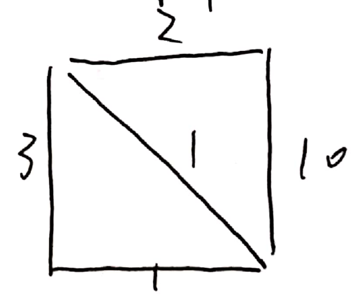
i=i-1 判断 w_i 的两个关联顶点 i', j' 若 tag(i')=tag(j')=true，跳到 step 1，~~若只有一个顶点 tag 为 true 另一个为 false~~ 否则到 step 2 => step 1

∴ 将标记的~~……~~为 false 的顶点的 tag 改成 true 并将该边加入 S_a 中，若所有顶点 i，tag(i)=true，则算法终止，否则回到 step 1 => step 2

a)



b)



c) 证明：按照上述加边的算法不会出现环，假设出现环 $w_0 w_1 w_2 \cdots w_n w_0$ 按照次序被加到 S_a 集合中

边 $w_0w_1, w_1w_2, w_2w_3, \cdots, w_{n-1}w_n, w_nw_0$

则必有 $w(w_0w_1) \ge w(w_1w_2) \ge \cdots \ge w(w_nw_0)$

这是不可能的，因为按照边加入的顺序 w_0, w_1 一定在 w_2 之前打上 true

同理 w_0 也在 w_n 之前被标记为 true 但 w_nw_0 表明 w_n 在 w_0 之前打上 true

二> 最小生成树实际上找的也是权值最大无环图 S_a，综上 n 为奇数

d) $w(S_a) \ge w(T_a)/2$ 设 T_a 生成~~最小~~树有 n+1 层，分别有 1, v_1, …, v_n 个顶点

·---- 第0层　第 i-1 层与 i 层之间有 v_i 条边，权值为 w_i

v_1 个顶　第1层　$w(S_a)$ 最小，则第0,1层 2,3层 ~~…~~ n-1,n层之间的边在 S_a

$w_1 + w_3 + \cdots + w_n \ge \frac{1}{2}(w_1 + w_2 + \cdots + w_n)$

v_n 个顶　第n层　因为 $w_2 > w_1$　$w_4 > w_2$ ……… 口

e) 算法在开头不需要排序