# 差分约束系统

需要求解一堆形如 $x_j - x_i \leq k$ 的不等式组。不等式 $x_j - x_i \leq k$ 代表了一条从 $x_i \to x_j$ 连一条边权为 $k$ 的边，系统的一个解对应了图的一个最短路。由于整个图不一定连通，因此通常加一个超级源点 $x_0$，边权为 $0$ 连向每个点。

显然若有一个解 $(x_1, x_2, \ldots, x_n)$，则 $(x_1 + d, x_2 + d, \ldots, x_n + d)$ 也是解。

## 最小解与最大解

注意上述过程其实是钦定了 $x_0 = 0$ 这一条件，同时可以证明最短路得到的所有变量 $x_i$ 都取到了该条件下能取到的最大值。

一个感性的理解：在用 $x_j - x_i \leq k$ 松弛 $x_j$ 的过程中，由于最短路的特性，$x_j - x_i$ 最大也只会取到 $k$，即使实际上给 $x_j$ 取更小一些的值也同样满足这个式子。即是说松弛过程取到了所有满足条件的值中尽可能大的，故所有解也是该条件下能取到的最大值。类似地，若将不等号取反跑最长路，得到的解是在 $x_0 = 0$ 下能取到的最小值。

## 实现中可能遇到的问题

1. 整个图不一定连通，因此要加一个超级源点。或者为了省事，也可以把所有点的 `dis` 初值都设为 $0$（但是仍然保持连通才能跑到每个点）；
2. 据说为了优化，可以用 DFS/BFS 模拟 SPFA，但是经过实测发现 DFS 还没有加了优化的 SPFA 快；
3. 图上可能有隐式的变量约束，如相邻两个变量的差绝对值小于 $1$ 之类的，需要额外加变量。

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
**https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:nikkukun:system_of_difference_constraints**

Last update: **2020/05/14 02:35**