问题概述

对偶图可以把最小割、最大流等问题转化为普通的最短路问题，从而明显降低算法的复杂度。

对偶图的定义

对偶图是与平面图相伴的一种图。对于给定的平面图$G = \langle E,V \rangle$设G的面为$F_1,F_2 \ldots F_e$当图G^*满足如下条件时，则图$G^* = \langle E^*,V^* \rangle$称为$G$的对偶图：

1. 对G的任意一个面F_i内部任选一点$v_i^* \in V^*$

2. 对F_i,F_j的每一条公共边界$e_x$$v_i^*$与$v_j^*$之间有一条边$e_x^*$与$e_x$交于一点。

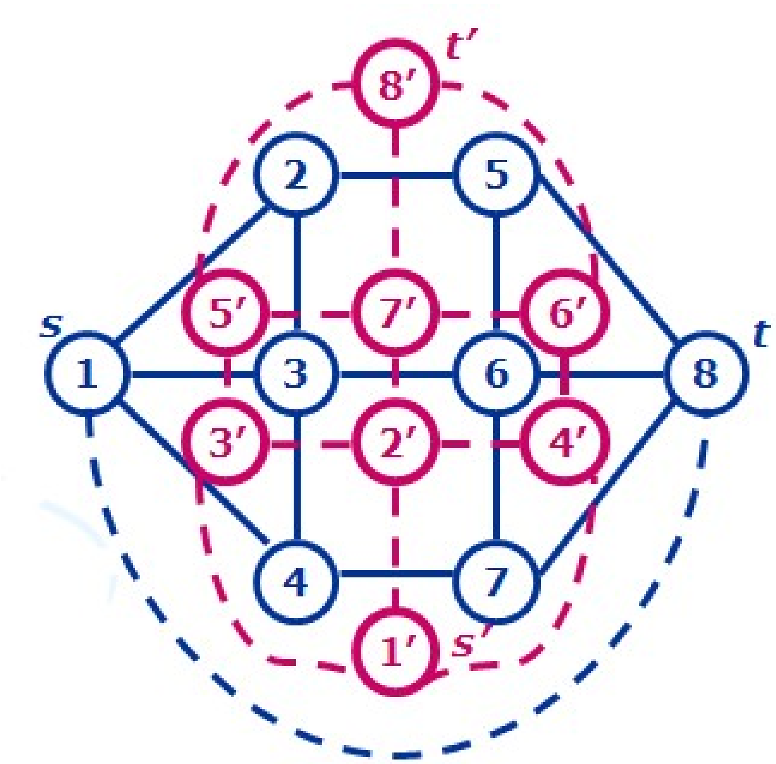
3.当且仅当e_x仅是一个面F_i的边界时v_i^*有一个自环与e_x相交。

例如下图中，白色的点和实线是原图点和边，黑色的点和虚线是对偶图的点和边。

通俗来说，对偶图就是把原来的图中的边围成的不同部分作为点，原来的图里的边两侧的部分连边，一般对偶图中的边和原图中与之相交的边权值相同。

对偶图的应用

很容易发现，在对偶图中经过一条边，与在原图中选择一条边相对应，所以我们发现，如果我们适当地决定起点和终点，我们就可以保证我们所走的路径是原图的一个“割”，联想到割的应用，我们发现求最小割除了最大流，还可以在对偶图上求最短路，比如下图



事实上，如果能把原图的对偶图建出并求最短路，那么时间复杂度几乎一定比最大流优秀，因为最大流的常规算法基本上都有$O(EV^2)$的复杂度，而最短路的复杂度可以达到稳定的$O(V \log V)$。但是，由于只知道图的点和边的信息是不足以建出对偶图的，所以大部分情况下我们仍然只能通过别的方法求出最小割。

例题——BZOJ1001狼抓兔子

题目大意

有一个$n \times m$的矩形，每个边有流量限制，问从$(1,1)$到达(n,m)最多有多少流量

解题思路

一眼看上去就想直接写最大流，但是由于这道题的数据范围较大，最大流算法必须要有不错的优化。又由于这道题给出的图非常规则，我们可以人工想出对偶图的样子，所以我们可以采用在对偶图上跑最短路的方式求出最小割，根据最大流最小割定理，我们得到的答案就是最大流

代码实现

自己写的太丑了，给一个博客