第 3 章 图　　论

3.1 路　　径

3.1.1 连通性

Network Attack

赛区/题库：Ural 1557　　　　　　　难度：★★★★☆

【题意概述】

给一个N个点M条边的无向连通图，求所有删除两条边使得图不连通的方案数。

数据范围：$1 \leqslant N \leqslant 2\,000$，$1 \leqslant M \leqslant 10^5$。

【算法分析】

对于一个图，如果切去了桥，该图必然不连通。那么我们要求的方案就可以分为两大类，第一类是至少一条边是桥，第二类是两条切掉的边都不是桥。算法框架如下：

（1）用深度优先搜索（DFS）求出所有的桥。

（2）缩点，即将桥连接的两点缩在一个块内。

（3）在缩点之后的图中找第二类方案。

（4）两类方案之和就是要求的方案数。

上面的算法要点就在于 3。为了叙述方便，我们规定下面的一些表示：

不妨从1号点开始 DFS 建树。设$A(i)$表示以i为根节点的子树中节点的集合，设$B(i)$表示点i的所有祖先组成的集合，$d(i)$表示节点i的深度，若i是j的父亲，则有$d(j) = d(i) + 1$，并令$d(1) = 1$。

令$C(A, B)$表示点集A中的点与点集B中的点相连的**边**的集合。

令$D(A, B)$表示点集A中的点与点集B中的点相连的**回边**的集合。

令$s(i) = \left|C\big(A(i), B(i)\big)\right|$，$t(i)$表示$D\big(A(i), B(i)\big)$中，$B(i)$中相关深度最大的节点的深度，也就是满足如下条件的$i$的某个祖先$j$的深度$d(j)$：存在$i$子树中的某个节点$x$，有$x$到$B(i)$中

某点j的回边，且$d(j)$最小。

证明第二类方案只有两种情况：

（1）$s(i)=2$时，$C\big(A(i),B(i)\big)$中的两条边。

（2）i是j的祖先，且$D\big(A(i),B(i)\big)=D\big(A(j),B(j)\big)$时（也就是说$A(i)\backslash A(j)$这部分没有到$A(1)\backslash A(i)$的回边，同时$A(j)$也没有到$A(i)\backslash A(j)$的回边），$i$与$j$各自父亲所连的边。

无向图的 DFS 遍历树没有横向边，我们得到：分割后的两部分不包含根的那部分要么是形如$A(i)$，要么形如$A(i)\backslash A(j)$。

如果分割后的两个集合形式为$A(i)$和$A(1)\backslash A(i)$，根据无向图的性质，要使两集合分开，则必须有$s(i)=2$，即$\big|C\big(A(i),B(i)\big)\big|=2$，取$C(A(i),B(i))$中两条边即为一种方案。

如果分割后的两个集合形式为：一个是$A(i)\backslash A(j)$，另一个是$A(1)\backslash\big(A(i)\backslash A(j)\big)$。则切割的两条边一定在$C\big(A(i),A(1)\backslash A(i)\big)$和$C\big(A(j),A(i)\backslash A(j)\big)$，$i$和$j$与各自父亲的边必然在这两个集合里面，所以这两个集合只能有这两条边。因此$D\big(A(i)\backslash A(j),A(1)\backslash A(i)\big)=\emptyset$且$D\big(A(j),A(i)\backslash A(j)\big)=\emptyset$，于是有$D\big(A(i),B(i)\big)=D\big(A(j),B(j)\big)$。

算法实现如下：

做一遍 DFS，利用子树的s和t值推出$s(i)$和$t(i)$。若$s(i)=1$则节点i与父亲的连边就是桥。若$s(i)=2$，则找到一种方案。若i与其父亲只有一条连边，则扫描$A(i)$中除了i的所有节点，若有j满足：

- j与其父亲只有一条连边；
- $s(j)=s(i)$；
- $t(j)<d(i)$。

则找到一种方案。这三个条件其实等价于$D\big(A(i),B(i)\big)=D\big(A(j),B(j)\big)$，整个算法的时间复杂度为$O(N^2+M)$。

【知识点】

图论

Synchrograph

赛区/题库：SGU 219　　　　　难度：★★★☆☆

【题意概述】

N个点M条有向边的图，每条边上有一个非负整数权值。一个点如果称为 active，当且仅当所有指向这个点的边的权值为正数。

下面会针对这个图进行一轮一轮的操作，每一轮会非确定的选择一个 active 的点，然后把这个点点燃。点燃这个点后，所有指向这个点的边会减一，所有这个点指出去的边会

加一。

每一轮，active 的点会随着操作而不断更新。而如果当前存在一个操作序列可以使得一个点成为 active 的，那么这个点称为有潜力的。如果经过任何操作序列一个点都会处于有潜力的状态，那么这个点就是 alive 的。现在要问图中哪些点不是 alive 的。

数据范围：$1 \leqslant N \leqslant 1\,000$，$1 \leqslant M \leqslant 5 \times 10^4$。

【算法分析】

容易证明，对于图中任何一个环，不管进行多少次操作，环上的权值总和都不会改变。而所有的边权值非负，所以发现导致点成为非 alive 的原因是存在边权都为0的环。可以发现，如果删掉所有权值非0的边，只剩下权值为0的边，那么构成的子图中，非单点的强连通分量中的点一定不是 alive 的，因为权值想要进入分量的话，必然从某一点进入，而又有分量内的其他点指向这个点。其次，非 alive 的点能到达的点同样不是 alive 的。因为非 alive 的点会在某一刻无法供应权值（即权值被减为 0），从而它所指向的点在有限次点燃后再也无法处于 active 状态。对于其余的点，沿反向弧前进会遇到两种情况。一是到达入度为零的点，从而保证了权值无限供应；二是形成环，而环上一定有权值非0的边，从而保证权值无限循环。两种情况无论哪一种都使得这个点为 alive 的。

所以这道题的算法应该是，对于权值为0的边构成的子图求一遍强连通分量，再进行搜索。

【注意事项】

数据中会有自环，自环的权值也可能会为0。

【知识点】

图论

3.1.2 欧拉路

Strange Graph

赛区/题库：SGU 156　　　　难度：★★★☆☆

【题意概述】

考察一个图$G = \langle V, E\rangle$。如果两个点有边相连，我们称之为邻居。在这个图中所有的边都是双向的。我们把v的邻居集合记为$N(v)$。记v的度为$\deg v$。

我们说一个图G奇怪是指如果图G连通，且对于每个顶点v满足如下情况：

（1）$\deg v \geqslant 2$（每个点都至少有两个邻居）。

（2）如果$\deg v = 2$那么与v相邻的两个点之间没有边。

（3）如果$\deg v > 2$那么存在一个u属于$N(v)$，且满足如下条件：

（a）$\deg u = 2$

（b）其他所有点w_1，w_2属于$N(v)\backslash\{u\}$是邻居。

给你一个这样的图，找到它的哈密尔顿回路。

数据范围：3≤N≤10^4，M≤10^5。

【算法分析】

这个图有它自己的特性。所有的度大于2的点都在团里，而度等于2的点把这些团连在一起。把团收缩成点，构造一个新图，新图中的点是原图中的团或者原图中度为2的点。且这个图满足任意两个代表团的点之间的路径必然经过代表原来度为2的点的点。而对于所有代表团的点，这个团中的每个点都连且仅连向一个原来度为2的点。于是对新图求欧拉回路，就可以还原出一条原图的哈密尔顿回路了。

【知识点】

图论

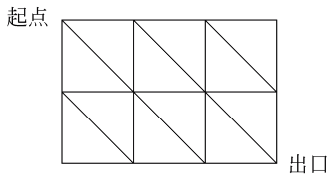
3.1.3 基本最短路

Animal Run

赛区/题库：Beijing 2006　　　　　难度：★★★★☆

【题意概述】

小动物们厌倦了动物园里的枯燥生活，于是决定展开一次大逃亡。它们偷得了动物园的地图，为一个$n \times m$网格状图，左上角的点是它们的起点，逃亡的终点是右下角的出口。图中连满水平和竖直的道路，还有成45°的斜向的道路，它们都是双向边。



不幸的是它们的逃亡计划被人类所看破，人们将阻止逃亡的发生。图中每条边上都有一个数值v_{ij}，表示多少个人在此就可以控制这条路。现在问最少需要多少人手，便可以将所有从起点到终点的路径控制住。

数据范围：$n,m \leqslant 100$， $v_{ij} \leqslant 10^6$。

【算法分析】

这道题很显然是求起点与终点间的最小割。但是图中光是点就有一百万个，最快的网络流算法也会超时，因此必须换一种思路。

这个图非常特别。把图中的三角形看做点，把两个相邻三角形之间的道路看作边，并且左侧和下面的三角形与左下方的外围相连，右侧和上方的三角形与右上方的外围相连，那么最小割等价于一条从左下到右上的最短路。点的数量有$2mn$，为百万级别的数，边数显然也是这个级别，用 Dijkstra+Heap 算法求出最短路即可。

【知识点】

图论

New Islands

赛区/题库：Tehran 2007　　　　难度：★★★★☆

【题意概述】

一张图，N个点，M条边。第i条边的权值为2^i。你需要构造一张新图，新图的边是原图中边的子集，并且满足新图中u,v的最短距离最多是原图中u,v最短距离的两倍。两点的最短距离定义为从u到v最少经过的边数。求满足条件的权值和最小的新图。

数据范围：$1 \leqslant N \leqslant 200$。

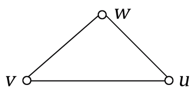
【算法分析】

因为$2^i > 2^0 + 2^1 + \cdots + 2^{i-1}$，所以只需要从大到小枚举边，如果能删除则删除，现在需要的就是如何快速判断一条边能否删除。

有一个显然的结论，就是如果原来所有距离为1的两点在删除某条边后距离最多为2，那么这条边就是可以删除的，也就是说，只需要考虑原来有边相连的两点在删除某条边后的距离。

删除一条边后会如何影响原来相连的两个点的距离呢？

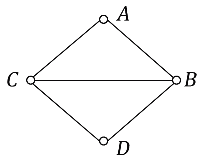
情况1：



$\exists w\big(w \neq u \,\&\, w \neq v \,\&\, E(w,u) \& \,E(w,v)\big)$，此时边$(u,v)$可以删除，否则不能删除。

是不是只需要判断u和v是否有与另外一个点相连呢？当然不是。

情况2：



考虑(B,C)能否被删，原来边(B,D)被删除了，那么(B,C)变成连接(B,D)且长度为2的一条路中的边，因此它能否被删除还需要考虑在每次删除后的新图中，对于已经被删除的边（u,v），统计u,v两个点能同时连向的第3个点的数目，若只剩一个，那么这一个点与u,v的连边就是不能删除的。

具体实现的话可以用$\text{tot}[i][j]$表示i和j可以同时连向的第3个点的数目，每删除一条边后更新tot就可以了。时间复杂度$O(MN)$。

【知识点】

图论

Recover Path

赛区/题库：SGU 515　　　　　难度：★★★☆☆

【题意概述】

给定一个N个点M条边的无向图，以及一些节点$v_1,v_2,\cdots,v_k$，你要选定起点v_s和终点v_t，使得从v_s到v_t的一条最短路径经过所有的$v_1,v_2,\cdots,v_k$。保证存在一条这样的路径。

数据范围：$N,M\leqslant 10^5$。

【算法分析】

首先以v_1为源点求一次单源最短路，选择一个离v_1最远的点v_a，然后再以v_a为源点求单源最短路。设$d(i)$为i离v_a的距离，构造一张新图：如果$d(i)+w(i,j)=d(j)$，那么连有向边(i,j)。很明显新图是一个有向无环图，我们在这个图上利用动态规划找一条包含$v_1,v_2,\cdots,v_k$中的点最多的路径，由于题目保证了有解，那么这条路径一定包含$v_1,v_2,\cdots,v_k$中的所有点。输出这条路径即可。

【知识点】

图论

Suffix-Replacement Grammars

赛区/题库：World Finals 2009　　　　难度：★★★★☆

【题意概述】

给出n个字符串后缀替换规则，每个由长度L不超过20的两个等长字符串A,B表示，表示若当前操作的字符串以A为后缀，则可以将其替换成B。现给出等长串S和T，问在S基础上最少需要替换后缀多少次才能变成T，需要首先判断是否有解。

比如，如果有4条规则$A \to B, AB \to BA, AA \to CC, CC \to BB$，那么把$AA$转换成$BB$可以：

（1）$AA \to AB$（使用$A \to B$规则）。

（2）$AB \to BA$（使用$AB \to BA$规则）。

（3）$BA \to BB$（使用$A \to B$规则）。

以上方案需要三次替换，但是实际上只需要两次即可：

（1）$AA \to CC$（使用$AA \to CC$规则）。

（2）$CC \to BB$（使用$CC \to BB$规则）。

数据范围：$n \leqslant 100$。

【算法分析】

这题很容易让人向动态规划方面考虑问题，但是状态的表示却让人非常头疼。搜索算法就更加不切实际，容易想象答案的数量级比较大（用类似分治的想法可以构造出）。

不妨从一个比较简单的问题考虑起，如果所有的规则字符串和起始目标字符串都等长，那么如何？实际上这就变成了非常简单显然的一道最短路的题目。再进一步分析，容易发现，可以直接忽略长度大于起始串长度的替换规则，换句话说，我们考虑长度为k的两个字符串时，只要考虑其他长度小于等于k的字符串就可以的。容易想象的是，在整个转换过程中只会有有限次长度为k的后缀（称这样的后缀为k-后缀）的替换，而这些替换实际上都存在于输入给定的规则中。两次k-后缀的替换之间，替换全部是由长度小于k的后缀替换组成，这样就转化成了一系列的子问题。

如果令S为所有规则的字符串和起始目标字符串的集合，S_k为S的所有长度为k的后缀的集合，那么要求的子问题就是S_{k-1}中任意两个串之间的替换距离。这样我们就可以利用子问题的答案构造S_k中仅用长度小于k的规则完成的替换距离。这时如果加上所有长度恰为k的替换规则，我们就完整地建立了替换距离矩阵，之后就可以使用 Floyd 算法了。复杂度为$O(n^3L)$。

【注意事项】

对于处理S_k集合，可以用字符串的 Hash 值代替，比较方便。

【知识点】

图论，字符串

3.1.4　有负权的最短路

Layout

赛区/题库：POJ 3169　　　　难度：★★☆☆☆

【题意概述】

有n头牛（编号1到n）在排成一队，编号较大者不能站在编号较小者前方。有一些限制条件，分为以下两种：

（1）$(x,y,maxd)$表示牛x和牛y的距离不能超过$maxd$，$x<y$；

（2）$(x,y,mind)$表示牛x和牛y的距离不能小于$mind$，$x<y$。

求方案使牛1和牛n的距离最大。

数据范围：$1\leqslant n\leqslant 1\,000$，$1\leqslant mind,\ maxd\leqslant 10^4$。

【算法分析】

差分约束系统。

此题可以进行转化，若设d_x为第x头牛的位置，则有若干个不等式，共分为以下 3 种：

（1）$d_x\leqslant d_{x+1}\rightarrow d_x\leqslant d_{x+1}+0$从$x+1$向$x$连长度为0的边。

（2）$d_y-d_x\leqslant maxd\rightarrow d_y\leqslant d_x+maxd$从$x$向$y$连长度为$maxd$的边。

（3）$d_y-d_x\geqslant mind\rightarrow d_x\leqslant d_y-mind$从$y$向$x$连长度为$-mind$的边。

用上述方法连好边后，这就是一个差分约束系统了，用最短路算法求解即可。若没有从1到n的路的话就说明牛1和牛n之间的距离可以取无穷大；若图中存在负环就说明无解；其余情况下1到n的最短路长度就是牛1和牛n的最大距离。

【知识点】

图论

Sightseeing Cows

赛区/题库：POJ 3621　　　　　　难度：★★★☆☆

【题意概述】

在一个n个点m条边的无向图中寻找一个环，使得这个环的点权和除以边权和最大。

数据范围：$n \leqslant 1\,000, m \leqslant 5\,000$。

【算法分析】

对于这类题目有一个很直接的想法就是二分答案ans后转化为判定问题，而转化后的判定问题大概就是要判断这张图里是否存在一个负（正）权环。

若答案比ans小，则对于所有的环都有

$$\frac{\sum_{i=1}^{c} v_i}{\sum_{i=1}^{c} e_i} \leqslant ans$$

即$\sum_{i=1}^{c}(ans \times e_i - v_i) \geqslant 0$。

于是可以重新设置边权，对于a与b两点之间的一条权值为w的无向边，在新图中变成两条有向边：从 a 向 b 连一条边权为$ans \times w - v_a$的边，从b向a连一条边权为$ans \times w - v_b$的边，之后在新图中寻找是否存在负权环即可。

关于使用 SPFA 判断负权环，只需要在 SPFA 的过程中判断是否有某个点的进队列次数大于等于n，若有，则存在负权环。

【知识点】

图论，二分

Word Rings

赛区/题库：CERC 2005　　　　难度：★★★★☆

【题意概述】

有N个单词，如果一个单词的最后两个字母和另一个单词的前两个字母相同，那么就可以看做这两个单词间连了一条有向边。

找出图中的一个环，使得单词平均长度最大。

数据范围：$N \leqslant 10^5$。

【算法分析】

我们把单词看成边，字母组合看成点。单词“$xxayy$”就是点“xx”向点“yy”连一

条长度为5的边。这样题目的规模变得小了许多，为一个点数至多为26^2的有向图。我们就可以用这类问题的一般方法——二分答案来做。

我们二分一个值L，为单词平均长度，然后把所有边权减去这个值。如果这个平均长度是合法的，我们一定可以在图中找到一个正权的环，这种情况我们应该把L增大，否则就把它变小。这样当二分的值趋于稳定的时候，我们就可以输出答案了。

在图中判有无正权环，通常用 SPFA 算法来做。开始将所有点设初始值 0，全都放进队列，然后用 SPFA 不断将点上的值往大的方向更新。如果存在一个正权环，那么沿着这个环将一直能更新下去，如此一来环上的点进队列的次数就会很多，因此判断当某个点进队列的次数超过总点数，说明存在正权环，否则 SPFA 算法可以正常结束。

【知识点】

图论，二分

3.2　匹　　配

3.2.1　二分图匹配

Double NP-hard

赛区/题库：UVa 10984　　　　难度：★★★☆☆

【题意概述】

求一个无向图$G(V,E)$的一个点集S，使得S同时是G的最大独立集，也是G的最小覆盖集。如果不存在则输出无解。

数据范围：$1 \leqslant N \leqslant 1000$，$1 \leqslant M \leqslant 10^5$。

【算法分析】

题目中涉及的名词比较多，下面逐个进行分析：

（1）有解的图中不含奇环。若图中存在奇环（长度为$2k+1, k>0$），环中的点最多有k个点在独立集中，而至少有$k+1$个点在覆盖集中，因此无解。

（2）有解的图可以划分为二分图，且不存在一种划分使得X部与Y部的大小不同。否则最大独立集大小不小于$\max\{|X|,|Y|\}$，最小覆盖集大小不超过$\min\{|X|,|Y|\}$，不可能满足要求。

（3）有解的图存在完美匹配。因为：在一个二分图中，最大匹配数=最小覆盖集大小，而一个最大独立集的补集就是一个最小覆盖集，要求的同时成为最大独立集和最小覆盖集的点集S，其大小必定为$N/2$（显然N为偶数）。于是，最大匹配数=最小覆盖集大小$=N/2$。