算法导论　第十四周作业　12 月 17 日　周四

PB18151866 龚小航

8.1（欧拉回路）强连通有向图 $G=(V,E)$ 中的一个欧拉回路是指一条遍历图 G 中每条边恰好一次的环路。不过，这条环路可以多次访问同一个结点。

(**a**). 证明：图 G 中有一条欧拉回路当且仅当对于图中的每个结点 v，有 $in\_degree(v)=out\_degree(v)$。

(**b**). 给出一个复杂度为 $O(E)$ 的算法来找出图 G 的一条欧拉回路。

解：分别分析

(**a**) 若某个节点的出度和入度不相等，那么若出度较大则对于该节点会无法遍历多的出边（进不来）；若入度较大则对于该节点会无法访问多的入边（出不去）。只有每个节点的出度和入度相等，欧拉回路的访问才是可转移的，即既不会困在一个点，也不会无法到达某条未访问的有向边的起始点。

(**b**) 只要从任意一个顶点开始，运行深度优先遍历算法(DFS)，把每次访问到的边打印出来即可。由于深度优先遍历时间复杂度为 $O(E)$ (每条边访问一次)，因此这么做符合题中所给的时间性能要求。

8.2 （套利交易）套利交易指的是使用货币汇率之间的差异来将一个单位的货币转换为多于一个单位的同种货币的行为。例如，假定 1 美元可以购买 49 印度卢比，1 印度卢比可以购买 2 日元，1 日元可以购买 0.0107 美元。那么通过在货币间进行转换，一个交易商可以从 1 美元开始，购买 $49*2*0.0107=1.0486$ 美元，从而获得 4.86% 的利润。

假定给定 n 种货币 $c_1,c_2,\dots,c_n$ 和一个 $n\times n$ 的汇率表 R，一个单位的 c_i 货币可以购买 $R[i,j]$ 单位的 c_j 货币。【思考题 24-3】

(**a**). 给出一个有效的算法来判断是否存在一个货币序列 $<c_{i_1},c_{i_2},\dots,c_{i_k}>$，使得：

$$R[i_1,i_2]\cdot R[i_2,i_3]\cdot\dots\cdot R[i_{k-1},i_k]\cdot R[i_k,i_1]>1$$

请分析算法运行时间。

(**b**). 给出一个有效算法来打印出这样一个序列（如果存在这样一种序列），分析算法的运行时间。

解：联系单源最短路径，应采用 Bellman – Ford 算法。

(**a**) 根据给出的形式，应该将乘法化为加法，故应该对其取对数。形式化为如下：

$$\lg R[i_1,i_2]+\lg R[i_2,i_3]+\dots+\lg R[i_{k-1},i_k]+\lg R[i_k,i_1]\ >\lg 1=0$$

即在一张完全图上找一 ju 个正环。而 Bellman – Ford 算法能较简单的报告图中存在的负环，因此可以对这个完全图中的每一条边权重取对数后再取负。因此最终求解形式如下：

$$-\lg R[i_1,i_2]-\lg R[i_2,i_3]-\dots-\lg R[i_{k-1},i_k]-\lg R[i_k,i_1]<-\lg 1=0$$

因此可以用 Bellman – Ford 算法在这个有向图上找出负环，如果存在负环，这个环上的节点系列就是满足题意的 $<c_{i_1},c_{i_2},\dots,c_{i_k}>$；若不存在负环则说明不存在这样的系列。

将所有边的权重处理一遍所需的时间为 $O(n^2)$，Bellman – Ford 算法的时间复杂度为 $O(VE)$，本题中的图是完全图，因此 $E=V(V-1)/2$。故总时间复杂度为 $O(V^3)=O(n^3)$

(**b**) 先将 Bellman – Ford 算法（发现负环时不退出）运行 V 次得到每个节点的 d 值，存储下来，再运行 V 次得到新的 d' 值。将它们相比较，负环就出现在 $d'<d$ 的节点中。在这些节点中任选一点，贪心的选出一个环即可。在这个过程中把访问的边打印出来。

8.3 假定在一个权重函数为 ω 的有向图上运行 $Johnson$ 算法。证明：如果图 G 包含一条权重为 0 的环路 c，那么对于环路 c 上的每条边 (u,v)，$\widehat{w}(u,v) = 0$。

解: $Johnson$ 算法将图中的边都重新赋予权重，并利用 Dijkstra 算法来求所有节点对之间的最短路径问题。由课本 409 页引理 25.1，可知重新赋予权重以后并不改变最短路径，也可以推出对于节点 v_0 到节点 v_k 的任意路径 p，都有 $\widehat{w}(p) = w(p) + h(v_0) - h(v_k)$. 在本题中， G 包含一条权重为 0 的环路 c，那么可以选这个环路上的任意一点，令其为 v_0 同时令其为 v_k，对于环路 c 而言必有 $\widehat{w}(c) = w(c) = 0$

另一方面，变换后的图中不含负边，因此对于环路 c 上的每一条边 $\widehat{w}(u,v) \geq 0$，而它们的和为 0，那么每一条都只能为 0。即 $\widehat{w}(u,v) = 0$

8.4 (最大流的更新) 设 $G = (V,E)$ 是一个源结点为 s 汇结点为 t 的流网络，其容量全部为整数值。假定我们已经给定 G 的一个最大流。

($\boldsymbol{a}$). 若将单条边 $(u,v) \in E$ 的容量增加 1 个单位，给出一个 $O(V+E)$ 时间的算法对最大流进行更新。

($\boldsymbol{b}$). 若将单条边 $(u,v) \in E$ 的容量减小 1 个单位，给出一个 $O(V+E)$ 时间的算法对最大流进行更新。

解：分别分析

($\boldsymbol{a}$) 将某条 E 中的边容量增加一个单位，等同于在残存网络上 (u,v) 增加 1. 因此只需要找增广路径，再运行一轮 $\mathrm{Ford-Fulkerson}$ 算法即可。由于给出的流本身就是最大流，那么增加 1 个容量的操作最多能让最大流增加 1，因此最多运行一次算法迭代，时间性能为 $O(V+E)$，即用 DFS/BFS 来找到一条增广路径。

($\boldsymbol{b}$) 将某条边容量减小一个单位，那么可以这么做：将已经知道的、减小之前的给出的最大流在经过的每条边上都令 $f' = f - 1$，先得到一个流量-1 的流 f'；将残存网络作相应的更新，重新运行一轮 $\mathrm{Ford-Fulkerson}$ 算法。由于 f' 最多增加 1，因此最多运行一轮迭代。和上一题相同，时间性能为 $O(V+E)$，即用 DFS/BFS 来找到一条增广路径。