3.4 图遍历

Problem 3.4.1 (DFS/BFS decompositions)
给定一个有向图和一个无向图，分别举例DFS和BFS的运作过程，各种边和点的着色。

Problem 3.4.2
请给出DFS算法基于栈的（非递归的）实现。

Problem 3.4.3
在有向图$G = (V, E)$上运行DFS算法，并且记录发现结点和离开结点的时间。对于有些边(u, v)可以观察到$f[v] < d[u]$。满足这样条件的边是下面哪种类型的边？（给出简要说明）

(i) Tree Edge (ii) Back Edge (iii) Forward Edge (iv) Cross Edge

Problem 3.4.4 (BFS对图中边的染色)
对于BFS，我们同样根据遍历的进行，把图中的边分为TE，BE，FE，CE。

(a) 对于有向图上的BFS，请证明以下性质：

i) 不存在FE。

ii) 对于TE (u, v)，我们有$v.d = u.d + 1$。

iii) 对于CE (u, v)，我们有$v.d \leq u.d + 1$。

iv) 对于BE (u, v)，我们有$0 \leq v.d \leq u.d$。

(b) 对于无向图上的BFS，请证明以下性质：

i) 不存在BE或FE。

ii) 对于TE (u, v)，我们有$v.d = u.d + 1$。

iii) 对于CE (u, v)，我们有$v.d = u.d$或者$v.d = u.d + 1$。

Problem 3.4.5
证明定理7.1[5]。

Problem 3.4.6
证明：一个有向图的condensation是无环的。

Problem 3.4.7
强连通分支算法中的两次深度优先搜索中，是否任意一个都可以（简单的）替换为一个广度优先搜索？请说明原因。

Problem 3.4.8
找出连通图的深度优先搜索树的根节点是割点的充要条件，并证明你的结论。

Problem 3.4.9
对于找割点算法，如果*back*被初始化为∞（或者$2(n + 1)$）而不是$dicoverTime[v]$时，算法是否还正确？说明你的理由。

[5]请参考Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析)第347页第三条

Problem 3.4.10
如果将找割点算法的test条件改为$back \geq discoverTime[v]$，那么二连通片算法是否还正确？如果正确，解释原因；如果不正确，给出反例。

Problem 3.4.11
无向图连通性和有向图连通性。

1. 证明在任意的连通无向图$G = (V, E)$中，存在这样一个顶点$v \in V$，删除它后G仍然连通。（提示：考虑G的DFS搜索树。）
2. 给出一个满足如下要求的强连通有向图$G = (V, E)$，对于任意的$v \in V$，将v从G中删除后，新产生的图不再强连通。
3. 在一个包含两个连通片的无向图中，通常可以通过添加一条边使得该无向图连通。给出一个满足如下要求的有向图，它包含两个强连通片，而只添加一条边不可以使得该图强连通。

Problem 3.4.12
请为如下任务设计一个线性时间[6]的算法：

输入： 一个连通的无向图G

问题： 是否可以从G中移除一条边，使G仍然保持连通？

能否将算法时间压缩在$O(|V|)$内？

Problem 3.4.13
给定无向图$G = (V, E)$，请给出判定是否可以为G中的边添加方向使得图中每个顶点的入度至少为1的算法，算法的时间复杂度为$O(V + E)$。若存在，算法需给出添加的方向。（提示：利用DFS算法。）

Problem 3.4.14
给定一个二叉树$T = (V, E)$（用邻接表表示），并指定其根顶点为$r \in V$。当在树T中从r到v的路径经过u时，u被称为是v的祖先。

我们想要通过对树进行预处理操作，使得形如“u是否是v的祖先？”的问题可以在常数时间内得到解答。预处理操作自身应该在线性时间内完成。请问预处理操作该如何进行？

Problem 3.4.15
给出一个线性时间算法，找出有向图中一条路径长度为奇数的环。

Problem 3.4.16
针对以下任务，给出一个线性时间算法：

输入： 一个有向无环图G

问题： G中是否存在一条有向路径，它恰好访问每个顶点一次？

Problem 3.4.17
假设v和w是同一个有向树中的两个不同顶点，并且它们之间没有祖先/子孙关系。证明存在第三个顶点c是它们的最小公共祖先，存在从c到v和w的路径，并且到这两个顶点的路径没有公共边。（提示：可以利用树的一个特性“从根顶点出发，有且只有唯一一条路径能到达树中的每一个顶点”。）

[6]对于图算法，我们称$O(n + m)$的算法为线性时间算法

Problem 3.4.18
下面两种定义在一个无向图的深度优先搜索树的顶点上的函数，都是在对于“两个顶点是否在图的同一个二连通片上”的问题尝试给出必要并/或充分条件。给出它们不成立的反例。

- 定义函数$old_1(x)$=顶点x的“最老”（即最接近根顶点）的祖先顶点，从x出发可以由多条tree edges和back edges到达；或者$old_1(x) = x$，当不存在这样的从x到达x的祖先顶点的路径时。证明对于v和w是否在同一个二连通片中的问题，$old_1(v) = old_1(w)$既不充分也不必要。
- 定义函数$old_2(x)$=顶点x的“最老”（即最接近根顶点）的祖先顶点，从x出发可以由多条tree edges和仅一条back edge到达；或者$old_2(x) = x$，当不存在这样的从x到达x的祖先顶点的路径时。证明对于v和w是否在同一个二连通片中的问题，$old_2(v) = old_2(w)$既不充分也不必要。

Problem 3.4.19
在一个有向无回路图$G = (V, E)$上，执行拓扑排序的另一种方法是重复地寻找一个入度为0的顶点，将该顶点输出，并将该顶点及其所有的出边从图中删除。解释如何来实现这一想法，才能使它的运行时间为$O(V + E)$。如果G中包含回路的话，这个算法在运行时会发生什么？

Problem 3.4.20
给定一个图$G = (V, E)$和一个起始点$s \in V$，BFS算法为V中的每个顶点u计算了从s到u的最短路径长度$d[u]$（路径上的边数）。在网络通信中通常还需要知道最短路径的数量。请修改BFS算法计算从s到图中每个顶点的最短路径的数量。这个问题的求解分为两个步骤：

(a) 首先在图G上运行标准的BFS算法。请说明如何利用BFS算法的执行结果产生一个新的图$G' = (V', E')$，其中$V' \subseteq V$并且$E' \subseteq E$，G'中的每条路径是G中起始顶点为s的最短路径，反过来，G中起始顶点为s的最短路径都在G' 中。

(b) 请说明怎样利用(a)中的结果来对V'中的每个顶点u求G'中从s到u的路径的数量，记为$c[u]$。(提示：考虑修改BFS算法)。

要求算法的时间复杂度是$O(V + E)$。

提示：在计算$c[u]$的时候，要特别注意。从一个结点到另一个结点的最短路径的数量可能达到顶点个数的指数级这个量级。这意味着你的算法不能逐条产生这样的路径。另外，你需要考虑怎样能够一次计数多条最短路径。

Problem 3.4.21
下面四个题目是和图中的环和路径的知识相关的。这里假设图是用邻接表表示的。（后面三个题目中只考虑无向图）

(a) 分别使用DFS和BFS两类图遍历框架解决有向图和无向图上的cycle detection问题。

(b) 算法老师声称他已经发现了判断图中是否包含环的复杂度为$O(V)$的算法，图是用邻接表来表示的。请证明算法老师提出的算法是不符合要求的。

提示：假设你已经有了算法老师的源代码。基于他的源代码，假设点的数目是一个任意值V，设计一个有$\Omega(V^2)$条边的有向图，如果他的算法不能遍历图中的所有边，你可以通过改变算法不能访问的边来使得算法执行产生错误的结果。

(c) 有向图$G = (V, E)$，如果对任意两个点$uv \in V$间都存在一条从u到v路径或者是从v到u的路径，就称G是一个半连通图。给定一个有向无环图(DAG)，请给出一个算法用来判断图是否是一个半连通图。

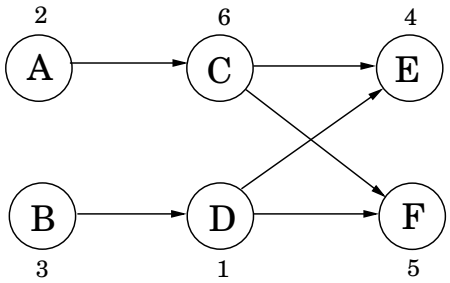
(d) 有向图$G=(V,E)$是一个半连通图。假设现在有一个半连通DAG G。请判断并简要说明G 中顶点的拓扑序是否是唯一的。

Problem 3.4.22

给定一个有向图，其中每个顶点$u \in V$都有一个与之关联的价格p_u，该价格是一个正整数。定义如下成本cost数组：对于每个顶点$u \in V$，

$$\text{cost}[u] = \text{从}u\text{可达的价格最低的顶点（也包含}u\text{自身）对应的价格。}$$

例如，在下图中（每个顶点的价格已标示出），顶点A、B、C、D、E、F各自的cost值分别为2、1、4、1、4、5。



设计一个算法，计算整个cost数组的取值（也就是计算每个顶点的cost值）。

(a) 给出一个针对有向无环图的线性时间算法。（提示：可以按照一种特定的顺序处理顶点。）

(b) 将上一问的算法推广至所有有向图的情况。（提示：回忆一下本章讲过的有向图的“两层”结构。）

Problem 3.4.23

Computopia市警察局为该市制定的交通规则规定所有的街道都是单行线。市长认为从市内的任一十字路口出发都要有一条合法的路线能够到达任一其他的十字路口，不过该主张还未得到反对派的支持。因此需要设计一个计算机程序，以确定市长的主张是否正确。然而，市内选举在即，所剩时间无几，只能允许设计一个线性时间算法。

1. 从图论的观点描述该问题，并说明为什么该问题能够在线性时间内解决。

2. 假定现在事实证明市长最初的观点是错误的。她随后又提出了一个妥协方案：如果你驱车从市政厅出发，沿着单行街道前进，那么不管你到哪里，总会有一条路线让你合法地驶回市政厅。将该妥协方案描述成一个图论问题，并详细证明能在线性时间内确定该方案的正确性。

Problem 3.4.24

课上已经讲解了找到图中桥的算法，我们可以把桥的概念扩展到有向图中。给定一个有向图$G=(V,E)$，如果在u和v之间不存在路径，那么称(v,u)是一个*span*。请给出一个判断DFS 算法产生的每条边是否是*span*的算法，算法的复杂度是$O(V+E)$。（注意Cross Edge和Forward Edge也有可能是*span*，但这里只要求查看DFS产生的树边）

提示：考虑修改课上所讲的二连通分片算法。

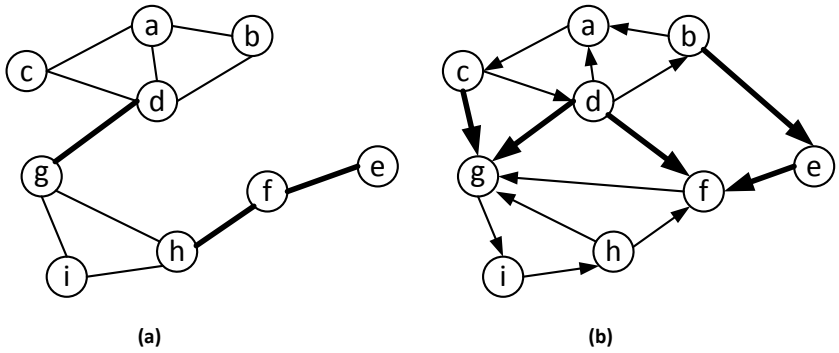


图 3.2: 图中的桥（粗线所示）

Problem 3.4.25
无穷路径。令$G=(V,E)$表示一个有向图，有一个"起始顶点"$s\in V$，一个"优等"顶点集合$V_G\subseteq V$和一个"劣等"顶点集合$V_B\subseteq V$。G的一条无穷路径p是一个含有无穷多个顶点的序列$v_0v_1v_2\cdots$，其中$v_i\in V$，并满足以下条件(1)$v_0=s$，(2)对于所有的$i\geq 0$，$(v_i,v_{i+1})\in E$。这意味着，p是G 中的一条从顶点s开始的无穷路径。由于集合V中顶点的数目是有限的，则G中的每条无穷路径一定访问了某些顶点无穷多次。

(a) 如果p是一条无穷路径，令$Inf(p)\subseteq V$表示满足以下条件的顶点的集合，它们在p中出现了无穷多次。证明$Inf(p)$是G中一个强连通片的子集。

(b) 给出一个算法，确定G中是否含有一条无穷路径。

(c) 给出一个算法，确定G中是否含有一条满足以下条件的无穷路径，它访问了V_G中的某个优等顶点无穷多次。

(d) 给出一个算法，确定G中是否含有一条满足以下条件的无穷路径，它访问了V_G中的某个优等顶点无穷多次，但却没有访问过V_B中的任何劣等顶点无穷多次。

Problem 3.4.26 (二部图)
给定无向图G，设计算法判定其是否为二部图(bipartite graph)？

Problem 3.4.27 (k度子图)
图$G=(V,E)$的诱导子图$H=(U,F)$满足$U\subseteq V, F\subseteq E\cap(U\times U)$。给定无向图$G=(V,E)$和整数$k$，找出$G$的最大诱导子图$H$，使得$H$满足每个顶点的度数至少为$k$，或者判定$H$不存在。算法的时间复杂度要求为$O(n+m)$.

Problem 3.4.28 (顶点间的"one-to-all"可达性问题)
给定一个有向图G，有n个顶点，m条边。

1. 请设计一个$O(m+n)$的算法来判断一个给定的顶点v能否到达图中所有的其他顶点。
2. 请设计一个$O(m+n)$的算法来判断图G中是否存在一个顶点，它可以到达图中所有的其他顶点。

Problem 3.4.29 (算法青年排队问题)

1. 将n个捣蛋的算法青年面朝前排成一队。输入m条信息"i恨j"。如果i恨j，则i不能排在j的后面某个位置，否则i会向j扔东西。请设计一个算法，在$O(m+n)$时间内将青年排成一队，或者判定不存在符合条件的排队方法。

2. 将n个捣蛋的算法青年排成多行。如果i恨j，则i所在的行号要小于j所在的行号。设计一个算法，找出所需要的最少行数。假如不存在满足要求的排法，请说明。

Problem 3.4.30

给定一个连通图$G=(V,E)$和一个顶点$u \in V$。假设已经找到了一棵以u作为根节点的深度优先搜索树T（包含图中的所有节点）。假设又找到了一棵以u作为根节点的广度优先搜索树$T'(T'=T)$。

(a) 如果G是一个无向图，请证明$G=T$。（也就是说，如果T既是G的深度优先搜索树又是G的广度优先搜索树，那么那么G中就不可能包含任何不在T中的边）。

(b) 如果G是一个有向图，(a)中的结论是否还成立，如果成立请给出证明，如果不成立请给出反例。

Problem 3.4.31

在2SAT问题中，给定一个子句的集合，其中每个子句是两个文字（一个文字是一个布尔变量或是一个布尔变量的取反）间的或（OR）操作。寻找一种方法，赋给每个变量true或者false，使得所有的子句都被满足。也就是说，使得在每个子句中至少存在一个取值为true的文字。例如，以下这个2SAT问题的实例：

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_4)$$

该实例有一个如下的满足赋值：分别将x_1、x_2、x_3和x_4赋值为true、false、false和true。

(a) 该2SAT问题实例是否存在其他的满足赋值？如果存在，找到它们。

(b) 给出一个含有4个变量的2SAT问题实例，使得该实例不存在满足赋值。

本题的意图在于引导我们找到一种求解2SAT问题的有效方法，该方法将原问题归约成一个在有向图中寻找强连通片的图论问题。给定一个含有n个变量和m个子句的2SAT 问题实例I，并按如下要求构建一个有向图$G_I=(V,E)$：

- G_I含有$2n$个顶点，每个顶点对应一个变量或变量的取反。
- G_I含有$2m$条边，对于实例I的每一个子句$(\alpha \vee \beta)$（其中α和β是文字），G_I 中都含有一条边从α的取反指向β，以及一条边从β的取反指向α。

注意到子句$(\alpha \vee \beta)$等价于蕴涵关系$\overline{\alpha} \Rightarrow \beta$或$\overline{\beta} \Rightarrow \alpha$。从这个意义上讲，$G_I$记录了实例$I$中的所有蕴涵关系。

(c) 构建本题给出的上述2SAT问题实例对应的有向图，并构建(b)问题中构建的2SAT实例对应的有向图。

(d) 证明如下结论：如果对于某个变量x，G_I中有一个同时包含x和$\overline{x}$的强连通片．那么实例I不存在满足赋值。

(e) 证明(d)的逆命题：即如果G_I中不存在一个强连通片中含有一个文字及该文字的取反，则该实例I一定是可满足的。（提示：按照如下方式为变量赋值：重复地选取G_I的一个汇点(sink)强连通片。将该汇点强连通片中的所有文字赋值为true，将其中所有文字的取反赋值为false，并删除这些文字。证明该过程最终将以发现一个满足赋值而终止。）

(f) 证明存在一个求解2SAT问题的线性时间算法。

Problem 3.4.32

假设现在有n只蝴蝶，每一只蝴蝶只可能属于两个品种A, B之一。现在通过下面的方法来将n个蝴蝶分为两类：判断任意两个蝴蝶i, j的种类是否相同，如果能够做出判断，就标记(i, j)这个判断是相同的或者不同的；如果无法做出判断，就称(i, j)为模糊的，并不进行任何标记。

最后得到了m个判断（不是相同就是不同，不包含模糊的判断），假设每个判断中涉及到的每一只蝴蝶确实只会是种类A或B，则称这m个判断是一致的。更具体地说，如果判断(i, j)为相同，那么i和j的种类相同并且不是A就是B；如果(i, j)判断为不同，那么$i(j)$来自A，$j(i)$来自B。

请给出能够在$O(m + n)$ 时间内判断出m个判断是否是一致的算法。

Problem 3.4.33

给定一个连通的无向图$G = (V, E)$，图中的每条边的权值为1或2。给出一个$O(V + E)$的计算G的最小生成树的算法。解释算法的正确性，并分析出其时间复杂度。（提示：可以利用BFS或DFS的变体来解决该问题。）