作业3:

练3.7:

解:(a): 由于线性时间,故考虑 BFS 或 DFS.

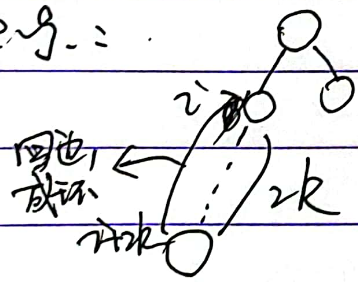
根据(b)问提示,选用 DFS,设起点为S,

从S开始进行DFS,将S标为颜色1,将与S直接相连的点标为颜色2,而下一轮中则将直接相连所有点颜色标为颜色1,重复直至结束。之后在遍历所有边,若存在一个边两端点颜色相同则不是二部图,则此无向图为二部图。

(b): ① "⇒": 图

若一个图为二部图,则其两个子集应分别包含偶数层与奇数层节点(此处"层"为DFS遍历的层级),~~而如果存在一个长度为奇数的环~~ 则对于其中一个子集,其中任意两个节点,设分别在2i与2i+2k层,则路径长度为2k,必为偶数,故不可能存在长度为奇数的环。(若存在,必有同一个子集中节点相连。)

e.g.:



→ 2i与2i+2k间成奇数环.

② "⇐":

若一个无向图不包含长度为奇数的环,则根据(a)中算法,可知用2种颜色就可以标记,所以这个无向图是一个~~二部图~~二部图

(c): ① 最多需3种颜色

② 仍采用类似(a)中的交替染色方法,仍使用DFS进行染色,设颜色分别为S1,S2,S3,交替染S1,S2,S3,则染到成环的那条回边的起点时,起点颜色与终点S肯定不同,~~因为路径长度~~

(c): ①: 最多需要3种颜色

②: 染色思路: 对于除唯一的环的那条链的起点,

使用颜色3进行染色, 而除此之外其他点与(a)中类似,

利用DFS交替的用颜色1.颜色2染色

3-11.

证: 此题思路应与课上讲的"去n-node tree has n-1 edge"

类似似。

先移除边e, 从e的顶点之一开始DFS, 若能到达

e的另个顶点, 则存在这样的环, 反之不存在

3-28:

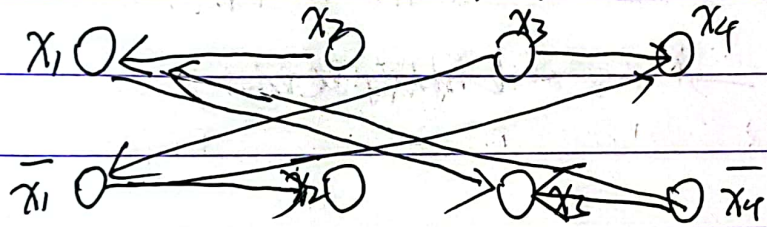
证:

(a): x_1= true x_2= true x_3= false x_4= ~~false~~ true

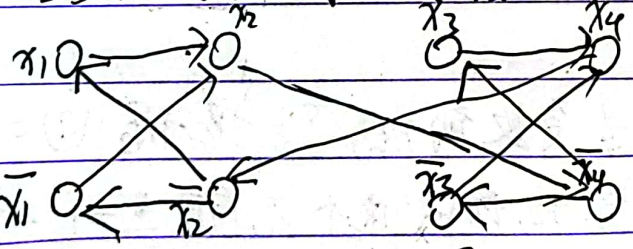
~~x_1= false x_2= true x_3= true~~ x_3= false x_4= false

(b): $(x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2) \wedge (x_3 \vee x_4) \wedge (\bar{x_3} \vee x_4) \wedge (\bar{x_2} \vee \bar{x_4})$

(c): (b)中2-SAT对应的有向图为:



(b)中2-SAT对应的有向图为:



(d): 证: 存在一个强SCC中同时包含x_i与$\bar{x_i}$

∵存在边 $x_i \to \bar{x_i}$ 与边 $\bar{x_i} \to x_i$

∴实例I中包含两个 $(\bar{x} \vee \bar{x})$ 与 $(x \vee x)$ 项

∴相互矛盾

∴此时实例I一定无满意赋值

e)：① 首先将 ~~实例~~ 实例 G2 生成的有向图中所有 SCC 均找出（使用 DFS，O(n)）。

②：将所有 SCC 均压缩为一个节点，则产生一个新的 DAG G_1（课上讲过的）

③：通过对 G_1 的转置图 G_1^T 进行拓扑排序（O(n)），找到 G_1 中一个 SCC

④ 将上述 SCC 中所有 ~~含~~ 含有的文字 x 取为 T，含有的所有 x̄ 对应 x 取为 F

⑤：将这个 SCC 从 G_1 中移除，重复 ③④ 至 G_1 为空

⑥：由 ~~前提~~ 前提：在一个 SCC 中均不含 x 与 x̄，故上述赋为 true 的 x 与赋为 F 的 x̄ 一定不冲突

⑦：综上，一定存在一种满意赋值，使 G2 可满足

f)：证明：该算法与 e) 中解法类似，但 ⑥ 步要改成：

⑥'：遍历所有 x（O(n)），检查是否有同时赋了 2 个值的情况（可对每个 x 设 2 个 attribute：setT 与 setF 来表示是否被设成 T/F，若均被设置，则矛盾）

⑦'：若不存在矛盾则输出对应满意赋值，否则无解

上述算法中 DFS，找 SCC 均为 O(n)，故总体为 O(n)

4-11：解：

~~①~~ 权值均为正，优先考虑 ~~Dijkstra~~ Dijkstra 算法。

① 对 ∀ u∈V，以其为起点，进行 Dijkstra 算法（其余各个点初始距离为 ∞）。

②：结束之后，维护一个最小长度变量 ans（初值为 ∞），

③：遍历 ∀ 边 e∈E，将其两个顶点的距离 ~~相加~~ 和相加

即 dist(u,v) + dist(v,u)，若 ans > dist 则 ans = dist

④：输出 ans，若 ans = ∞，则无环

时间分析

在 $E < V^2$ 时，若以无序 array 来实现堆，则一次 Dijkstra 为 $O(V^2)$，故总时间为：（最差）

$O(V \cdot V^2 + E) = O(V^3 + V^2) = O(|V|^3)$

4.12：

解：设 $e = (u, v)$（即两个端点分别为 u, v）

∵ 为无向图

∴ 可分别考虑 $u \to v$ 和 $v \to u$ 两种。

①：对于 $u \to v$，从 v 开始使用 Dijkstra 算法，之后将 $|e|$ 与 dist(v, u) 相加，得 ans1。

②：对于 $v \to u$ 方向，从 u 开始进行 Dijkstra 算法，之后将 $|e|$ 与 dist(u, v) 相加，得 ans2。

③：比较 ans1 与 ans2，若均为 ∞，则没有包含 e 的环，反之则输出 ans1 与 ans2 中较小者

时间复杂度：

以无序 array 实现堆，则单次 Dijkstra 为 $O(|V|^2)$

故总体为：$O(2|V|^2 + 1) = O(|V|^2)$

4.16：

解：(a)：∵ 是完全二叉树

∴ 设顶点 j 位于第 k 层从左向右第 m 个，则有 $j = 2^k - 1 + m$

∴ 其父顶点应位于 k-1 层，第 $\lceil m/2 \rceil$ 处

即 $p = 2^{k-1} - 1 + m/2 = \lfloor \frac{j}{2} \rfloor$

∴当 j 为父顶点时 其孩子顶点 k 应满足 $\lfloor k/2 \rfloor = j$

∴ $k = 2j$ 或 $2j+1$

∴综上 j 的父顶点在 $\lfloor \frac{j}{2} \rfloor$ 处，孩子在 $2j$ 与 $2j+1$ 处

(b)：设某个顶点 j，且其父子点 的末端超过总数 n

设 j 位于第 k 层从左向右第 m 个，

则 $j = \frac{d^{k}-1}{d-1} + m$.

∴其父节点位于 (k-1) 层 $\lceil \frac{m}{d} \rceil$ 个的位置

∴ $j_{p} = \frac{d^{k-1}-1}{d-1} + \lfloor \frac{m+d}{d} \rfloor = \frac{d^{k}-1}{d-1} + \lfloor \frac{m+d}{d} \rfloor$

∴ $j_p = \lfloor \frac{j-2+d}{d} \rfloor$

∴ $\lfloor \frac{k-2+d}{d} \rfloor = j$

~~∴ $k = dj+1, -d, dj-d, \cdots dj+1$~~

∴孩子位置为 $dj+k$，$k \in [2-d, 1]$（代入 d=2，与上一问同，本应该没问题）

∴ { 父节点位置 $j_p = \lfloor \frac{j+d-2}{d} \rfloor$

孩子位置 $dj+k$，$k \in [2-d, 1]$

(c)：证明：

①最坏情况为每一个新结点都需不断向下调整

②对于一个序号为 i 的结点，其最终会下移到最底层，

其下层深度为：$\Theta(\log n - \log i) = \log(\frac{n}{i})$

∴最坏为 $\Theta(\sum_{i=1}^{n} \log(\frac{n}{i})) = \log \frac{n^n}{n!}$

由斯特林公式 $\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}(\frac{n}{e})^n} = 1$

∴ $\sum_{i=1}^{n} \log(\frac{n}{i}) = \log \frac{n^n}{n!} \approx \log \frac{n^n}{\sqrt{2\pi n}\frac{n^n}{e^n}} = \log \frac{e^n}{\sqrt{2\pi n}}$

$= n - \frac{1}{2}\log(2\pi n)$

∴ $O(\sum_{i=1}^{n} \log(\frac{n}{i})) = O(n)$

综上可得，即 make heap 最坏时间复杂度为 $O(n)$

③ 最差情况对应输入：为从左向右单调递减的序列

④ ① 将 bubble up 中 $p=\lceil i/2 \rceil \Rightarrow p=\lceil (i+d-2)/d \rceil$

② 将 minchild 中：$\text{argmin}\{key(ch(i)) : 2i \le j \le \min(ch, 2i)\}$

改为 $\text{argmin}\{key(ch(i)) : d(i-1)+d \le j \le \min(|h|, d(i+1))\}$.