计算机学院《算法设计与分析》
(2019 年秋季学期)

第四次作业参考答案

1 对下面的每个描述，请判断其是正确或错误，或无法判断正误。对于你判为错误的描述，请说明它为什么是错的。(每小题 5 分，共 20 分)

1. NP-$hard \subseteq NP$；
2. 对某问题 $X \in NP$ 而言，若可以证明规约式 3-$SAT \leq_p X$，则 $X \in NPC$；
3. $P \neq NP$；
4. 所有 NP 完全问题均无法在多项式时间内被解决。

解：

1. 错误，NP-$hard$ 问题未必多项式时间内可验证；
2. 正确；
3. 无法判断；
4. 无法判断。

2 最小生成树性质的证明 (20 分)

令 $G = (V, E)$ 为一个带权无向连通图，且其每条边的权值都不相同。请证明：G 的任何一棵最小生成树都会包含权值最小的那条边。[注：必须从头开始证明，即是说，证明过程中不可使用 MST 引理，即关于“安全边 (*Saft Edge*)”的引理，且证明不可建立在 $Kruskal$ 算法或 $Prim$ 算法成立的基础上。]

解：

假设权值最小的那条边 (记为 $e = (u, v)$) 不在该图的最小生成树 (记为 T) 中，如果我们把 e 加入到 T 中，就会导致新的子图 $T' = T \cup \{e\}$ 中包含一个环。令 e' 是该环中除了 e 之外的任意一条边，因为 e 是原图中权值最小的边，所以 $w(e')$ 一定大于 $w(e)$，我们令 $T'' = T \cup \{e\} - \{e'\}$，即从子图 T' 中删去 e' 这条边，那么 T'' 仍然是原图的一棵生成树，并且有：

$$w(T'') = W(T) + w(e) - w(e') < w(T)$$

这和 T 是原图的最小生成树矛盾，因此命题得证。

3 影响传播问题 (20 分)

给定一个有向图 $G = (V, E)$，其中每个点 v 的权值为 $t(v)$，每条有向边 (u, v) 的权值为 $w(u, v)$。现将一点 r 标记为活跃，随后 r 会向他的邻居传播影响。影响传播的方式如下：每条边 (u, v) 表示如果 u 为活跃节点，那么 u 会对 v 传播大小为这条边的权值 $(w(u, v))$ 的影响；而每当一个节点 v 收到的影响超过其权值 $t(v)$ 后，该节点也将变为活跃节点。请设计算法计算图 G 中最终会有几个活跃节点, 并分析其时间复杂度。

解：

解决该问题的主要思路是构建一个活跃节点的队列。最开始，队列中仅有一个节点 r。

…… (2 分)

接下来，只要队列非空，就从中取出一个节点，并更新该节点对它邻居节点的影响。

…… (8 分)

随后检查它的每一个邻居节点，如果某节点 v 其当前状态为不活跃，但积累的活跃值已经超过了该节点的上限 $t(v)$，那么就将其标记为活跃并放入队列中。 …… (8 分)

该算法通过 BFS 很容易实现。最坏情况下需要遍历所有点和所有边，因此时间复杂度为 $O(|V| + |E|)$。 …… (2 分)

算法伪代码如 Algorithm 1 所示。

Algorithm 1 $Inf(G, t, w, r)$

Input: 有向图 G，点权 t，边权 w, 初始活跃结点 r
Output: 活跃的结点数

```
Q ← {r}
sum ← 1
while Non-Empty(Q) do
    u ← Q.front()
    Q.pop()
    for each (u, v) ∈ E do
        t(v) ← t(v) − w(u, v)
        if t(v) < 0 then
            Q.push(v)
            sum ← sum + 1
        end if
    end for
end while
return sum
```

4 骨牌覆盖问题 (20 分)

给定一个大小为 $n \times m$ 的棋盘，其中某些位置被损坏了，如图 1 所示。棋盘的信息通过矩阵 R 给出，$R[i][j] = 0$ 表示该位置是完好的，$R[i][j] = 1$ 表示该位置被损坏了。现请你使用大小为 1×2 的骨牌来覆盖整个棋盘（如图 2 所示，注意：必须**恰好**覆盖该棋盘，换言之，所有损坏的地方以及超出棋盘边界的地方均不能被骨牌覆盖）。请你设计算法判断是否能恰好覆盖整个棋盘并证明该算法的正确性，给出该算法的时间复杂度。(时间复杂度为 $O(n^2m^2)$ 的算法可得满分)

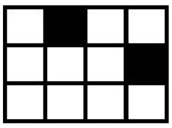


图 1: 一个 3×4 的棋盘，有两个位置被损坏了

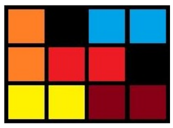


图 2: 该棋盘可以被恰好覆盖

解：

将棋盘中每一个没有被损坏的格子视为一个节点，每个节点和它上下左右没有被损坏的格子连一条无向边，这样我们可以得到一个无向图。可以证明，该无向图为二分图。证明过程如下：

将棋盘第 i 行第 j 列的格子对应的节点编号为 (i,j)，并将节点按照 $i+j$ 的奇偶性进行分类。令

$$A=\{(i,j)|i+j\text{为奇数}\},B=\{(i,j)|i+j\text{为偶数}\}$$

可以发现所有的边连接的两个节点一定是一个属于集合 A，另一个属于集合 B。故该图为二分图。

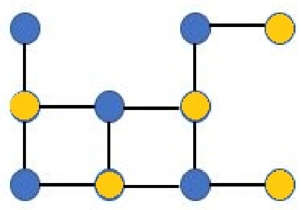


图 3: 将棋盘转化为一个二分图

而每个 1×2 的骨牌会覆盖该二分图中相连的两个节点。这样，判断该棋盘是否可以被恰好覆盖，等价于判断等式

$$\text{图中的最大匹配}=\text{节点数}/2$$

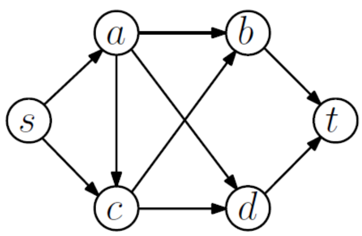
是否成立。

求二分图的最大匹配可以使用匈牙利算法，其时间复杂度为 $O(|V|\times|E|)$，在本题中，图的节点个数与边数均为 $O(nm)$，故时间复杂度为 $O(n^2m^2)$。

5 路径统计问题 (20 分)

给定一个有向无环图 $G=(V,E)$ 以及图上两点 s,t。请设计算法计算图 G 中从 s 到 t 的路径数量并分析其时间复杂度。

例如，对于如下所示的包含 6 个点的图，从 s 到 t 共有 6 条路径。（仅需求出路径的数量）



1. $s\to a\to b\to t$;
2. $s\to a\to c\to b\to t$;
3. $s\to a\to c\to d\to t$;
4. $s\to a\to d\to t$;
5. $s\to c\to b\to t$;
6. $s\to c\to d\to t$;

解：

定义状态 $num[v]$ 表示从点 s 到点 v 的路径个数。 …… (3 分)

枚举所有指向 v 的边 $(u,v)\in E$，则有如下递归式：

$$num[v]=\sum_{u:(u,v)\in E}num[u]$$

…… (8 分)

初始化仅需将 $num[s]$ 为 1，我们要求的即为 $num[t]$。 …… (1 分)

需要注意的是动态规划的计算顺序，在计算 $num[v]$ 之前，需要保证对于所有 v 的前驱结点 u，$num[u]$ 均已被计算过了。因此，需要将原图进行拓扑排序，之后按照拓扑序进行枚举计算。…… (5 分)

拓扑排序和在图上进行动态规划的时间复杂度均为 $O(|V|+|E|)$，因此，总的时间复杂度为 $O(|V|+|E|)$。 …… (3 分)

算法伪代码如 Algorithm 2 所示。

Algorithm 2 $numPath(G, s, t)$

Input: 有向无环图 G，图上两点 s, t

Output: s 到 t 的路径个数

对图进行拓扑排序，得到结点的拓扑序 $a[1], a[2], \cdots, a[|V|]$
$num[s] \leftarrow 0$
for $i \leftarrow 1$ **to** $|V|$ **do**
    for each $(u, a[i]) \in E$ **do**
        $num[a[i]] \leftarrow num[a[i]] + num[u]$
    end for
end for
return $num[t]$