分布式算法作业

EX2.1.分析在同步和异步模型下，convergecast 算法的时间复杂性。

答：

同步：在 convergecast 算法的每个容许执行中，树中每个高为 t 的子树，根节点在第 t 轮里收到所有孩子节点的 msg。

证明：

（1）第一轮时，高为 1 的子树根节点收到来自叶子节点的 msg

（2）假设第 t 轮时，高为 t 的子树根节点收到所有孩子节点的 msg,则第 t-1 轮开始时，这些节点向父节点发送 msg。

所以第 t+1 轮时高为 t+1 的子树根节点收到所有孩子的 msg。

所以 d 为树高，则同步 convergecast 算法的时间复杂度为 O(d)。

异步：在 convergecast 算法的每个容许执行中，树中每个高为 t 的子树，根节点在时刻 t 收到所有孩子节点的 msg。

证明：

（1）当 t=1 时，初始时各叶子节点的 msg 处在从叶子节点到其父节点的传输中，由异步模型的时间复杂性定义知，各叶子节点的父节点（即高度为 1 的子树的树根）至多在时刻 1 收到某子节点的 msg.

（2）设 $P_i \in \{树高于t的子树根节点\}$, P_j 是 P_i 的子节点，则 P_j 的树高至多是 t-1 的子树，

由归纳假设知，P_j 至多在 t-1 时刻收到 P_j 子节点的所有 msg，所以，P_i 至多在时刻 t 收到所有孩子的 msg。

所以 d 为树高时，存在一个时间复杂度为 d 的异步 convergecast 算法。

EX2.2.证明在引理 2.6 中，一个处理器在图 G 中是从 Pr 可达的，当且仅当它的 parent 变量归纳曾被赋值过。

证明：

（1）因为 G 是 parent 和 children 变量所确定的静止图，所以只有收到 M，即算法第 5 行被执行。又因为是容许执行，所以算法第 7 行被执行，即 parent 变量被赋值。

（2）若 parent 变量被赋值，则算法的第 7 行被执行，又因为是容许执行，所以第 5 行必被执行过，即收到过 M，因此 Pr 到该节点是可达的。

EX2.3.证明 Alg2.3 构造一刻 Pr 为根的 DFS 树。

证明：

（1）反证。假设某节点在 G 中从 Pr 不可达，因网络是联通的，若存在两个相邻节点 Pi 和 Pj，使得 Pj 在 G 中是从 Pi 可达的，而 Pi 是不可达，因为 G 里面一结点从 Pr 可达当且仅当它曾经设置过自己的 parent 变量，由算法可知 Pi 仍保留在 Pj 的 unexplored 数组中，即 unexplored 数组不能为空，因此 Pi 被搜索到之前算法不会终止，因为是容许执行，所以 Pi 最终比被搜索到，并被 parent 赋值，因此得出矛盾，故图中所有节点从 Pr 均可到达。

（2）证明无环。Pi1，Pi2，......Pj，若 Pi 是 Pj 的孩子，则 Pi 在 Pj 第一次收到 M 之后，

第一次收到 M，因为每个处理器在该环上是下一个处理器的反素，这就意味这 Pi 和 Pj 在第一次收到 M 之后收到 M，矛盾，所以无环。
由算法可知，在任意时刻算法中仅有 1 个 msg 在传输或被传输或被接收，当 unexplored 未空时，必向 unexplored 中某一个结点发送 M，只有当前结点的邻居都已被搜索之后，才向上一层发送 parent 消息，因此算法构造的是一棵深度优先树。

EX2.4.证明 Alg2.3 的时间复杂度为 O(m)。

证明：
（1）在同步模式下，由算法可知，算法中当且仅当有一个消息在传输，且仅发往一个处理器结点，除根节点外，所有处理器都是收到消息后才被激活，所以不存在多个处理器在同一轮发送消息的情况，因此消息复杂度和时间复杂度一致时 O(m)。
（2）在异步模式下，在一个时刻内至多有一个消息在传输，因此时间复杂度也与消息复杂度一致，而对任一边可能传输的消息最多有 4 个，即 2 个 M 和 2 个回应 M 的消息<parent>和<reject>，所以时间复杂度为 O(m)。
综上，时间复杂度为 O(m)。

EX2.5.修改 Alg2.3 获得一新算法，使构造 DFS 树的时间复杂度为 O(n)。

```
code for processor Pi,0<=i<=n-1
Wr parent:init nil;
   children:init φ
   unexplored:init all the neighbors of Pi;

   upon receiving no msg:
    if(i=r) and (parent=nil) then
    {
       parent:=i;
       send <already> to all processors in unexplored;
       for all Pj in unexplored;
       remove Pj from unexplored;
       send M to Pj;
    }
    upon receiving <already> from Pj;
    {
       remove Pj from unexplored;
    }
    upon receiving <M> from Pj;
    {
       parent :=j;
       remove Pj from unexplored;
       if (unexplored !=empty) then
       {
```

```
        for all Pk in unexplored;
        remove Pk from unexplored;
        send M to Pk;
        send <already> to all prcessors in unexplored;
    }else send <parent> to parent;
}else send <reject> to Pj;
upon receiving <parent> or <reject> from neigh our Pj;
if receiving <parent> then add j to children;
if unexplored is empty then
{
   if parent !=i then send <parent> to parent;
   terminate;
}else
{
   for all Pk in unexplored;
   remove Pk from unexplored;
   send M to PK;
}
```

EX3.1 证明同步系统中不存在匿名的一致性的选举算法

证：在匿名系统中，所有处理器没有统一标识，由由于一致性算法不知道处理器数目，故每轮时间内所以处理器状态均相同，经过若干次执行后，仍具有相同状态，其中既不存在选中状态也不存在非选中状态，所以与选举算法不矛盾。

EX3.2.证明异步系统中不存在匿名选择选举算法

证：由于同步算法是异步算法的一个特例，即每轮时间相同传输距离相同，同步系统中不存在匿名选举算法故意不系统也不存在。

EX3.9.若将环 R_n^{rev} 划分为长度为 j（j 位 2 的幂）的连续片段，则所以这些片段的次序是等价的。

证明：任意整数 P 可表示为如下形式：$P=\sum_{i=1}^{m} a_i 2^{i-1}, m=\lg n P \in [a,n-1]$，则

$rev(p)=\sum_{i=1}^{m} a_i 2^{m-i}$，设 P，Q 在一个片段上，且设这两个片段是相邻的，根据模加法有

$P_1=P+1, Q_1=Q+1$，其中 1 为片段长度，设 $l=2^k$，则 $P=\sum_{i=1}^{m} a_i 2^{i-1}$，$Q=\sum_{i=1}^{m} b_i 2^{i-1}$。

因为 P,Q 在同一个片段上，所以 $|P-Q|<1=2^k$，所以存在 $r\in\{1,k\}$，使得 $a_r != b_r$，否则

$\left|P-Q\right|>=2^{k}=1$，这与 P,,Q 在同一个片段矛盾。

设 S=min(r)，则根据 rev(p>rev(Q))的表示方法得，sign=(rev(P)-rev(Q))=sign(as-bs),而

$P_1=P_{t1},Q_1=Q_{t1}+1$，可得，P 与 P_1，的前 k 位是想通的，Q 与 Q_1 的前 k 位是相通的。

有 0<=S<=k 得，sign=(rev(P_1)-rev(Q_1))=sign(as-bs)，则这两个相邻片段的次序是等价的，根据等价关系传递性，可得所以片段次序相等。