对Paxos算法的理解：

Paxos是一种分布式一致性算法，简单说就是在多台机器之间通过一致性协商过程，让所有机器保存同样的数据。客户端访问任一台机器，得到的结果都会是一样的。

这种一致性的存储十分重要。大型系统常常需要存储一些被频繁修改和访问的数据。这种存储常常成为一种关键单点，如果单点故障，会影响整个系统的服务。因此，必须将单点变成分布式系统，也就是说，将数据同时存储在多台机器上，即使部分机器故障，依然不影响整个系统的服务。将数据存储多台机器，就必然产生一致性的问题。多台机器同时的改动，如何达成一致，同步到所有的机器。Paxos就是解决这种一致性问题的基础算法。

Paxos将角色分为提议者，接受者，学习者。学习者不参与协商过程，可以先忽略。算法流程如下：

1，提议者任意发起提议。所有的提议都要分配一个不重复的自增的ID。这里提议者应该理解为发起提议的机器，因为同一个机器，也可以是接受者。同时多个机器可以同时发起提议，但是提议ID必须是全局自增的。

2，接受者要保存接受过的最大的ID，并且只处理大于等于当前ID的提议。如果提议ID合法，则返回本机器已同意接受的提议，没有则返回提议为空。

3，提议者如果收到过半数的机器返回，并且返回值都为空，提议者认为提议可行，向所有机器广播提交提议。如果返回值里面有非空，提议者放弃自身的提议，选择非空里面ID最大的提交。如果返回不足半数，跳到第一步，重新提议。这里体现了两个协商原则，首先，机器需要有过半数返回才能协商成功。其次，提议是先到先得，只要有其他提议先被接受了，当前提议者只能服从。

4，接受者接收到提交的提议，同样地只处理ID较大的提议。如果提议合法，则接受提议。

5，提议者如果收到过半数的机器接受提议，则提议成功。否则，返回第一步重试。

如果3和5环节失败了，提议者都会回到1环节获得新的ID重试，重试是很有必要的，保证了在网络不稳定的情况下，系统也能最大程度的保证达成一致。