已知条件：

1. 用户一次性可购票1-5张
2. 座位总共4个区域，每个区域最前排50个，最后排100个，隔排递增2个

假设条件：

1. 每个区域总共可以为25/26行（25行则为只有一行100，26为2行100），我们假设是26行。
2. 由假设条件1可以算出每个区域的座位总数为：（50+100）\* 26 ／2 = 1950

产品抽象：

这里已知条件是一次性购票限制，故选取用户行为维度（即每一次为一个购买进程）。随机分配先随机分配区域，然后随机分配行，最后分配列。最终将座位固定。

数据结构：

如图所示：



后续产品落地优化：

1. 搭建web站来运行系统
2. 用户信息存储数据库，购买前需登录，前置校验
3. 购买约束仍旧为session 维度
4. server端票数据的存储考虑到不为传统意义上的关系型数据，而且数据量小唯一，所有用户共用，可以使用redis来进行对象存储
5. 对于共享资源（也就是票数据）需要建立事物锁来确保数据的准确性
6. 算法边界条件的优化处理（如某区域票不足的情况下，需要考虑跨区域拼单）

交付：

见压缩包，解压后执行node index.js即可