模拟启明电梯运行系统学习过程

通过学习电梯运行系统，收获良多。主要有以下几点。

1·了解了时间调度算法：综合考虑时间效益、实时性和稳定性，选择了LOOK-SCAN算法，并加上了自己的一些想法，在请求时间一样的情况下，选择方向一致的。这些可以见程序。

2·在码程序的过程中从用户的角度考虑：最开始，我是先打了一个深搜的程序（注：程序已经丢失）（小白打起来真的很费劲），目标是节省电费———吝啬的物业———且先不说，程序本身存在的逻辑漏洞。基本打完了时，我觉得这个程序对用户极度不友好———有点心虚———于是，我又打了一个最短等待时间的程序，深刻体贴用户，感觉要好一些，但是同样有逻辑漏洞，等会再讲。

3·学习了算法：在打第一个程序时，之前学了深搜和贪心结合，只会记录结果，而不会记录过程。这个电梯运行对我而言确实是一个很大的挑战，我尝试着用返回值来判断是否需要记录路径。在打第二个程序时，我仍旧选择了搜索算法，并学习回溯（状态量很多，回溯起来很麻烦）。在打第三个程序，也就是最终版，由于电梯路径的实时性，我放弃了搜索算法并尽可能地减少递归中的变量，增加全局变量，来使得程序显得简洁而有更强的连贯性。同时，将程序大体分为三个部分：预处理区、控制区和执行区，更加具有条理性，并且在隔一段时间后也能很快地熟悉代码，补完代码。在程序设计上的难度是第一个>第二个>第三个。

4·逻辑漏洞和算法的深入理解（一把泪）：第一个程序，想的是节省电费，要真正实现节省电费就得知道全部呼叫时间，而这在实际中是不可能的。如果只是局部的最优，在这种情形下并不能保证全局最优，反而可能会导致资源的浪费。再加上极度糟糕的用户体验，可以说这个程序一无是处QAQ。第二个程序，在代码中虽然遍历各种可能路径时加上了呼叫时间的限制，让程序看似是在对未来的呼叫一无所知的情况下运行，但实际上，由于搜索的算法特性，在遍历了所有路径后选择了其中一条。这本身就是已经掌握了所有信息，再来做出判断，在实际中也是不可能实现的QAQ，但至少还存在一定价值，能够实现局部的最优（win！！！）。第三个程序，学乖了，只考虑当前情况，使当前路径唯一可选择，具有实际应用价值。