第一轮： OA waiting list, + f1: 如何优化 + f2: 桌子能做的人数完全match party人数 +

f3: 小餐馆要桌子完全match party人数，大餐馆不需要 + f4: 等待时间大于30分钟就不考虑以上规则

地理面经重复率挺高的coding是wait list 开始是一个list 有ts 有人数，找第一个人数小于桌子的party  
follow up 1， 另外一个饭店必须是桌子正好等于party人数才可以入座  
follow up 2， 把上面两个function整合，楼主写了个if，不过感觉是在问strategy pattern

第三轮：coding，华人小姐姐，特别nice，题太简单了..就是餐馆有个座位，有各种group顾客选座那个；很快做完就欢声笑语了

她的需求linkedlist就行

https://www.1point3acres.com/bbs/thread-558194-1-1.html

code轮是个萝莉风格的小姐姐，叫克莉丝汀，题目是wait list，大致就是input两参数一个是list 一个是桌子size k。 就是一个排队的队伍里，怎么把人分到桌子size 为k上去。 地里老题， 当时就开始巴拉巴拉pq解法，后来一写发现压根不需要，sort一下，然后for找到比k小的人群就行了。非常easy level。 followup是另外一种分配方式该怎么写，忘了是啥，但是也很简单，就是加个if 就完事了。

题目如下：  
已知有一张open table，和一个waiting list [{size, name, queue\_time},{...},...]  
要求返回等候最久且party size <= table size 的 party name  
并且把这个party从waiting list中删除  
  
Follow up:  
已知有两种餐厅，大餐厅和小餐厅  
大餐厅就是上面提到的那种  
小餐厅优先安排人多的party入座，只有party size == table size的时候才会安排party入座（返回party name + waiting list中删除）  
但是当有一个party的queue\_time >= 30min时，优先安排这个party  
当没有party size == table size或者等待超过30min的party size > table size时，小餐厅和大餐厅一样，都是优先queue\_time  
一些条件：  
当waiting list为空，或没有party 满足要求时，输出errmsg  
不会有queue time相同的两个party  
。。好像还有一些 我忘了 反正他最后左一句右一句的 时间不够 没有全部写完  
感觉不算是算法题吧？更偏OOD一点？反正不是地理的高频题，应该吧？

A: LZ我有个问题 input的 list of party 是按照时间排序好的吗？我觉得如果用双链表或者单链表来做这题的话，如果不是 sort好的，好像就失去链表的意义了

input是没有排序的，需要自己写comparator

地里原题 waitlist。  
直接用一个像LRU那样的双向链表就解决了  
follow up 是 解决饭馆只想4人桌坐4个人，队里即使只有2人的party也不给安排。  
跑了几个测试  
问了他几个问题就结束了

A:

链接找不到了  
输入是list of party, 每个party是用dict表示，然后你写几个函数比如moveWL, addPartyToWL这样就行。  
自己定义数据结构，你也可以试试其他结构。  
他的意思是先写出个能用的，没必要特别复杂。之前地里有个用PQ的挂了。。尽量别PQ了。

我的意思就写个deque，deque里放party，然后有桌子就从头开始找，找到第一个可以坐的就remove。但是这样删的操作比较慢  
所以写个LRU那样的双向链表可以加速

<https://www.1point3acres.com/bbs/thread-542165-2-1.html>

给的题目是yelp的 waiting list， 主要就是让你自己写一下比较基本的waiting list 系统。  
有点像简单的OOD 的题目  
要求就是会有很多party进来waiting list 等有空桌子坐下来吃饭，每个party可能有的field是party的size， name， 还有等的多久时间。  
然后table 可能就是有 status，比如目前这个桌子是空的还是说已经有人坐下了。  
然后有空的桌子的话，你就从等最久的那个party看起，如果他们能做的下，就坐下，不然就看等的时间第二久的party看看能不能坐下，以此类推。  
我用的就是自己写的party class 然后comparator写的就是直接比较他们的time。  
小姐姐要求还是蛮简单的，我一开始写的很复杂就是比如同时有很多桌子空了， 该怎么安排之类的。  
然后小姐姐说我们就假定只有一个空桌子，然后有三四个party在等的那种情况就好了。  
然后我就修改了一下 最后test case run 了几个也都过了。求onsite~~~~~~

A: 不太适合用priority queue把，毕竟就是先等的party 要优先考虑，这种情况还是用linked list存party就好了。

这种情况用linkedlist的，如果头的那个party坐不下就直接往后一个找，直到找到合适的就delete from linked list， 这样会比我这种用pq的好，我用pq是poll出来的存着，最后再重新insert进去。

第二轮面试是一个waitlist系统，怎么安排餐厅最先seat谁  
筛选规则是只要空出来的位置能够坐得下，就优先排等位时间长的party