主要数据结构（下划线表示核心算法相关）：

任务（taskID，类型，deadline，startTime，地点坐标，指定送餐员列表，状态）：类型分为取餐和送餐（送餐失败后的返还直接修改送餐地址为商家），startTime表示如果提前到达则需等待至startTime，状态为进行中／待分配。进行中的任务送餐员列表中有且仅有一个人。核心算法不应考虑任务类型，仅考虑待分配任务，进行中的任务仅用来预期送餐员可进行下一项任务时的时间和地点。

送餐失败后暂时不需要返回，有可能需要稍后重新配送或者直接丢弃

提前到或者提前完成该任务的时候，不需要等待startime，而是及时更新任务的状态，触发新一轮计算

关于状态，从核心算法的角度来看，对任务来说，不管成功或者失败都应该视为完成。我之前的考虑是，对于每个配送任务，只有 未分配，已分配，已取餐，已完成 四个状态，其中只有在 未分配和已分配 时可以被重新分配，一旦取餐，则只有这个司机可以去配送（如果出现意外，则又作为新任务进入待分配列表）

任务依赖关系（二元组（taskID1，taskID2））表明task2必须在task1完成之后才可由完成task1的送餐员进行。每一个订单产生取餐和送餐两个任务，送餐依赖于取餐。

送餐员（driverID，坐标，正在进行中的任务ID，工作时间，状态，携带的订单列表）：状态分为任务进行中／空闲／不可用。核心算法仅需要知道送餐员可用的时间和地点。携带订单等信息产生的影响通过任务的指定送餐员列表反映。

订单（orderID，商家ID，客户坐标，菜品列表）：无需核心算法考虑，可能需要技术人员考虑的是同一个订单的菜品是否可拆分的问题，建议不可拆分，如有需要可新建订单。（不需要拆分）

地图（某些坐标之间的距离、时间、建议路线）：需要跟地图模块进行交互。算法应该考虑估计时间的不确定性，对于时间的变动应该健壮／稳定？

核心算法的返回：每个送餐员一个顺序任务列表。满足任务依赖关系和指定送餐员的限制。首先保证按时完成尽可能多的任务，其次让分配到任务的司机数目尽可能少。司机下班时不能携带订单。（如果订单取消，下班后一次送回就行了）

p.s. 测试的时候可以写一个仿真模块。

功能实现：

1. 商家在下单前需要得到一个送餐员到达的估计时间

可通过加入虚拟订单来实现，虚拟订单只设置取餐任务，按照平均的响应时间设置deadline？

2. 送餐员到达餐厅的时候餐馆没有准备好餐

如果当前时间未超过取餐任务的startTime，则原地等待无需汇报。否则通知调度人员，将送餐员的状态置为空闲，取餐任务状态置为待分配，如有必要重新设置取餐任务startTime和deadline。如果调度算法仍然分配这个任务给送餐员，则送餐员在原地等待。其实有一点顾虑：送餐员如果不按规定汇报可能会影响分配策略。（具体的业务需求是这样的：商家下单前会知道大概的取餐时间，司机去晚了没事，去早了需要等到承诺的取餐时间，如果到了给商家通知的取餐时间商家还没做好的话，最多等N分钟，N为预设参数，司机直接离开，这一个任务就算是 完成 了）

3. 送餐员在路上出现意外（抛锚或者有事故）

将送餐员状态置为不可用，将其携带的订单的取餐地址置为送餐员接头位置的坐标。

4. 送餐员配送失败（客户不在家）

通知调度将送餐地址修改为商家。（不需要，直接设置为该任务完成，但是如果要再次分配的话，只能分配给这个司机）

5. 调度人员重新分配任务给别的送餐员

直接修改送餐员和任务的相关数据

6. 商家要求重新配送？不太理解

直接修改订单相关数据（类似过了取餐时间但商家没有准备好，客人不要或者联系不上之类的，后来商家又联系好了，需要重新排配送，技术上可以理解为一个新任务就可以了）

其他：

对于运算部分，最常见的场景应该是 新来一个任务 （包含取餐地址，送达地址），怎样获得分配最优解，一般来说，我们可以通过 Google Maps获得多个地点的Best Route以及到每个节点的时间，那就需要考虑下面很多问题：

Google Maps的API的调用频率有限制；另外设计地图模块进行处理

当然司机如果正在开始配送（已经取到餐）则不应打乱当前未配送的任务，最早只能插入到下一个还没有开始配送的任务（可以是已经取到的）；前面设计中假设的是当前任务都不可被调度算法改变，包括取餐任务（正在取餐的路上）。

某司机已开始的任务不能再调整之后超过一个最长等待时间；每一个任务都包含取和送两部分，这两部分可以分别整合到已有的司机那里；不太理解（参考一开始关于状态的说明）

如果某个任务delay的时间太久，需要考虑取消他未开始的任务（未取）；每隔一段时间后台调用一次调度算法，算法会根据目前的状态重新分配任务。

某些场景配送失败，但是餐还在这个司机身上，则重新分配的时候只能选择这个司机（除非是有事故）；设计中通过设置指定送餐员列表实现。

很多条件的判断标准可能是一些我们设定好的参数，并且参数应该便于调整；需要考虑当前司机是否在工作时间，看看前面的设计行不。是否在工作时间

触发调度算法的事件（等你接口写出来再完善）：

新单

新送餐员

订单取消

送餐员下班

取到餐

送到餐

取餐任务状态变更（直接取消／重置／等待？）

送餐任务状态变更（返还？／待定？）

调度员任务指定

Current Task (pick up / delivery)；Next Target address; deadline time

Coming Task 1 (pick up / delivery); Target address; expired time

Coming Task 2 (pick up / delivery); Target address; expired time

注意，每个司机都会有一个类似上面的任务列表

这个列表是否有必要显示给司机？