/\*basic demand:

现在要求实现的基本问题：

1.机器人类型：人形、轮形和机器狗形

(1)对于每个机器人来说，vi1表示在边上的移动速度，vi2表示在楼梯上的移动速度

(2)对于每个机器人来说，有robotskill代表该机器人能完成的工作类型

2.Tasks

(1)坐标={ord,stair},stair代表在第几层，ord代表在stair层数上标号为ord节点的位置

(2)Task包含skill，（对简单基础模型而言） 我们只考虑有一个skill类型，即不需要多个机器人组合完成该项目（后期优化内容）

(3)Task包含内容，即从初始坐标前往目标坐标{ord2,stair2}

(4)在倪老师给出的基本框架内，我们考虑最原始的情况，即任务一个一个处理，不要求做到最优任务分配（后期优化），每个task包含时间time

3. 电梯、楼梯坐标{ord,stair}

(1)电梯和楼梯：电梯的移动速度非匀速，楼梯的移动速度是匀速的，且不同类型的机器人的移动速度恒定并且不变。

(2)对于电梯，不同类型机器人的移动速度是固定的，但是电梯不是匀速的，意味着从1到3楼的边长小于1到2楼边长的两倍，将3维图转化为二维图时候需要注意。

4.难点：如何处理冲突问题：

(1)根据倪老师给我的介绍，边是一个双向车道，每个车道仅能有一个机器人意味着不同时间段内边会有占用的情况，（我还不知道有什么好的解决方案）

(2)电梯：电梯内容量有限（似乎仅能容下一个机器人？） 会有电梯运行冲突。

5.实现方式：

(1)目前解决方法：贪心+最短路（就是不知道冲突如何解决）？

//目前建图有个难点：如何通过楼梯和电梯去把三维图转化为二维图，会出现环和稠密图的问题

Input:

Robot[]:{type,skill,ord,floor,speed1,speed2/\*on stair\*/}

Tasks[]:{target\_skill,ord,floor,target\_ord,target\_floor,time/\*arrial\*/}

lift[]:{ord,speed}

stair[]:{ord,floor}

Edge[]:{u,v,length}

Output:

/\*for each task\*/

solution[]:{robot\_num,route}

Model://这部分是把三维图转化为二维图，建立跨楼层的边

fake\_code:

Tag[ord\*ord]//记录每条边是否被占用，内包含time表示结束时间，默认为0（空闲）

Tag\_lift//记录电梯是否被占用，tag\_lift.start表示开始占用的时间,tag\_lift.end 表示结束时间，默认tag\_lift.start=tag\_lift.end=-1表示无占用

Current\_time→0

for task∈Tasks do:

for r∈robots do:

if r.skill==task.skill://贪心：

Get r a shortest solution to {r.ord,r.floor}：

//迪杰斯特拉or SPFA? 但是要考虑的事情有点多，需要大改动

//including two ways: lift or stair,calculate twice

//Avoid any roads occupied:

对于每一条边：

If current\_time+arrival\_time>Tag[road\_num]→allow to use

else →now allow to use

Refresh tag\_lift(if necessary)

Refresh Tag[ord\*ord]→arrival\_time+road\_length+current\_time

Continue;