**更改的所有的代码按照要求都在myAgents.py里面**

思路简单来说是这样的

1.因为是多智能体协作问题，所有的多智能体都会去寻找距离最近的食物(使用bfs算法)，因为可能存在不同的智能体选取相同的一个食物作为目标，因而我在MyAgent类中构建了一个全局变量target列表，通过所有的食物判断是否为目标时通过

*if state not in* target *and* self.food[x][y]==*True*:

约束食物只能进入target列表一次，因而使得不会存在不同智能体选取一个食物。

2.如果说所有的智能体每次都进行一次最小路径的判断的话，太过于浪费思考时间，因而我们设置为每个智能体进行了一次最近食物判断之后，下次决策会按照上次的最小路径进行运行直到路径为0(每个智能体维护一个path的列表)，而后再进行一次最近食物判断

3.同时我们设置了游戏的开始变量和结束变量，这主要是去约束当最后如果只剩1个食物，而有2个智能体，此时1个智能体会去吃掉最后的食物，而另一个智能体会判断为游戏结束，呈现静止状态

4.为了让不同的智能体负责不同的区域(也就是分区的思想)，我们通过调整不同的智能体上下左右入队的顺序来使得不同的智能体尽可能地分开，定义了四种不同的入队顺序，同时为了减少思考的时间，在智能体中维护一个开始的标志变量，规定只有刚开始的时候才会定义dfs算法不同方向的入队顺序，而后会一直沿用这种入队顺序，同时，为了维护这种入队顺序，myAgents.py类重新定义了

*def* breadthFirstSearch(*problem*,*direction*):

*def* getSuccessors(*self*, *state*,*direction*):

两个函数

5.最后一点点说明，关于为什么不更改代价函数让不同智能体选择去往智能体多的地方的代价更高，其实是这么写过，但是因为思考时间过长消耗的Scores过多，反而得不偿失