地图构建 ：

Pathfinding.Heuristics.init\_joint\_policy\_graphs （ starts,ends ）｛

输入参数是所有agents的起始点和终点集合，要求类型为list ，并且stars和ends长度相等，再将starts和ends一一对应并拆开，形成对应每个agents的start和end，调用BFS（）函数单独为agent搜索路径。中间使用assert\_in\_bounds判断是否越过地图边界，地图边界表示为 x\_bound和y\_bound

没有返回值，其结果保存在pathfinding.Heuristics.dijkstra.graphs中，

｝

碰撞避免：

Mstar\_OD.search（ starts,ends ）{

根据上方的地图，找出所有agent都移动一步时的总代价。

}