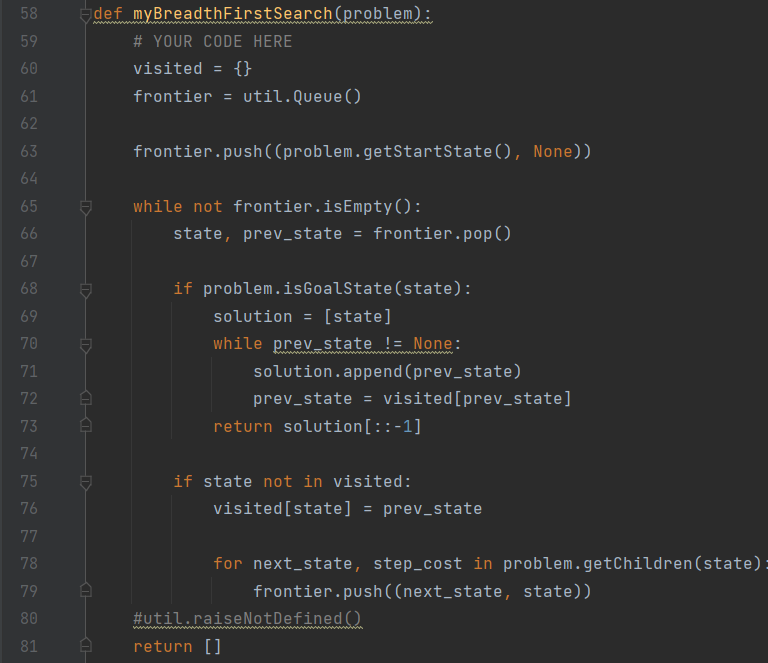
# 人工智能基础 LAB1 实验

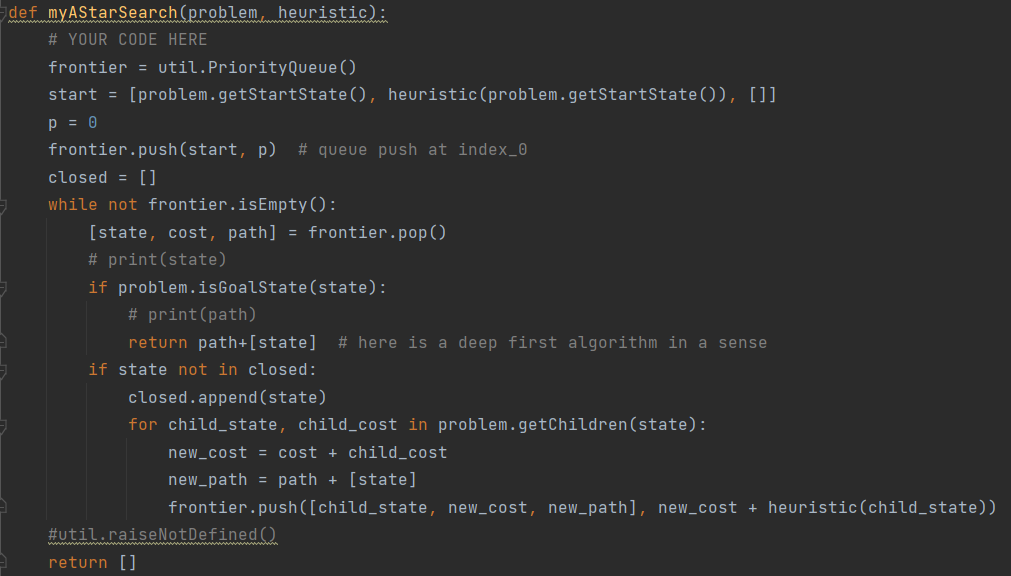
PB18111699魏钊

**BFS:**



只需修改深度优先搜索的代码即可，两者的区别在于，深度优先搜索使用栈存储，广度优先搜索使用队列存储。

**AStar:**



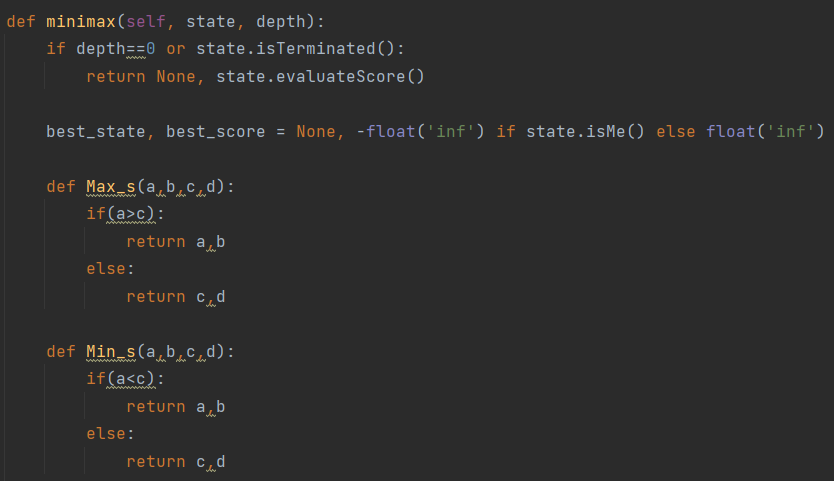
**AStar算法使用优先队列存储，start为初始状态及信息。**

**将start优先度设为0，压进队列，然后开始循环，判断是否为目标状态，若是，则返回路径；若不是，判断当前状态是否访问过，若没有访问过则存进closed。**

**然后将当前状态的后代状态及相关信息压进队列，队列不为空则继续循环。**

**经过动画演示对比发现：深度优先搜索红色最少，Astar搜索次之，广度优先搜索最多，即三者搜索过的路径数目。**

**Minmax：**



**当前深度为0或已经为终止状态时返回。**

**定义了Max\_s和Min\_s用于比较分数大小，返回分数及相关状态。**

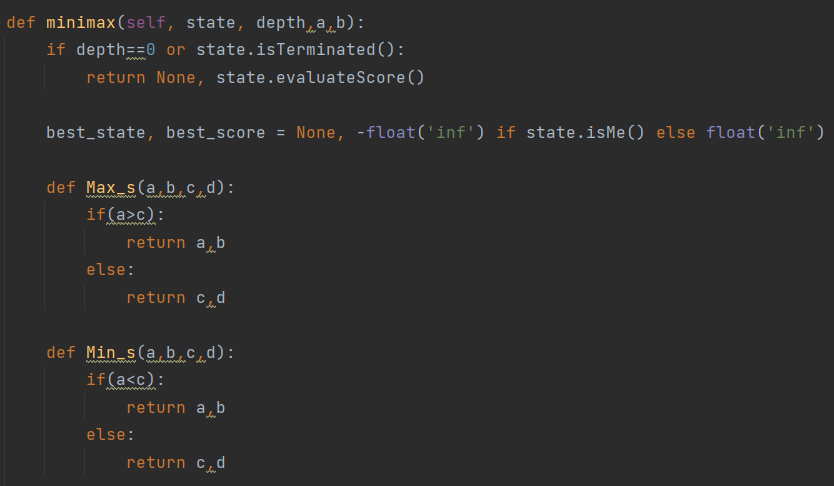


首先判断状态是否为Agent，若是，则其孩子为Ghost，需要从孩子分数（最小分）中选出最大的。

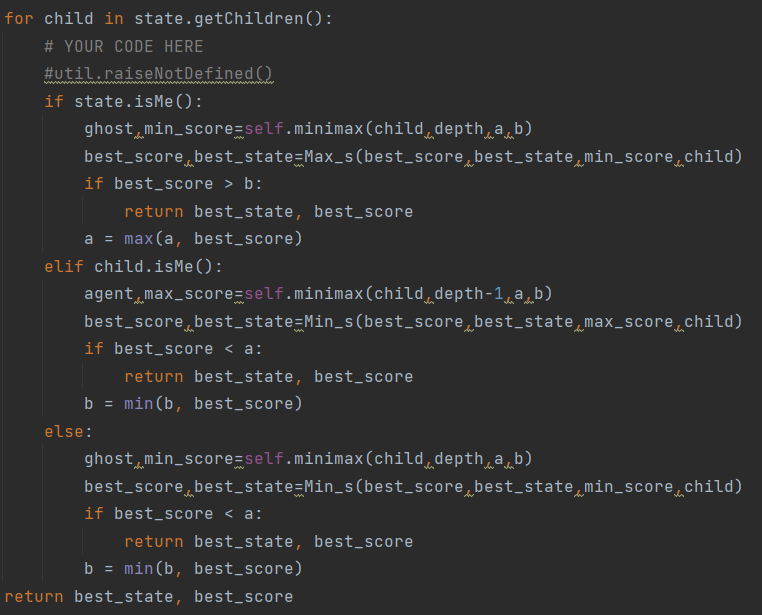
若当前状态不是Agent（即Ghost）需判断其孩子状态是否为Agent,若是，则需要从孩子分数（最大分）中选出最小。（此时需深度减一）

最后，当前状态和孩子状态均为Ghost，需从最小分数中选出最小。

**AlphaBeta：**



首先和Minmax类似的，定义函数并加上端点值a,b。



大部分代码和Minmax相同，还有加上每次的剪枝操作，如上图有三种，当复合if的剪枝条件时，返回，否则修改端点值。