A\*算法：

定义search node：包含board，移动到该board的移动次数，以及先前（一个？）的board。

1. 把初始node（初始board，move=0，空的predecessor node）插入到一个优先队列中。
2. 从优先队列中弹出一个priority最小的node，然后往优先队列中插入它的neighbor nodes（不包含predecessor node）。
3. 重复2直到从优先队列中弹出的为目标node。

Priority的定义：hamming priority或者Manhattan priority。似乎后者好一点。

Manhattan priority等于所有blocks的Manhattan距离（block到目标位置的水平+垂直距离之和）之和，再加上到达该node的move数量。

Hamming priority 等于位于错误位置的block的数目，加上到达该node的move数量。

避免Manhattan距离的重复计算：在构建一个node的时候就计算好其Manhattan距离，存在一个实例变量中。

检验不可解的情况：若初始board交换两个block就能得到goal board，那么它不可解。