Best Path

**This method is a brute force search that looks at every choice until it runs out or guarantees that it has found the best path already. It can be good for situations that require some amount of brute force, but otherwise it is very slow and it is also difficult to make the helper functions 100% bug free.**

**Helper Functions:**

posmoves(path) – generate all possible moves given a current path.

isbest(path) – determine if a path is better than the current bestpath. If bestpath is empty, any path should be better.

isdone(path) – determine if a path has reached its goal.

isfailed(path) – determine if a path should no longer be explored. Don’t worry about checking if there are no more possible moves from this path.

**def** solve**(**x**):**

bestpath **=** **[]**

**def** posmoves**(**path**):**

**pass**

**def** isbest**(**path**):**

**if** **not** bestpath**:** **return** **True**

**pass**

**def** isdone**(**path**):**

**pass**

# no need to check for dead ends where no more moves are possible here

**def** isfailed**(**path**):**

**pass**

# recursive findpath

**def** findpath**(**path**):**

**nonlocal** bestpath

**for** move **in** posmoves**(**path**):**

newpath **=** path**.**copy**()**

newpath**.**append**(**move**)**

**if** isfailed**(**newpath**):** **return**

**elif** isdone**(**newpath**):**

**if** isbest**(**newpath**):**

bestpath **=** newpath

# keep looking once a path is complete

#findpath(newpath)

**else:**

findpath**(**newpath**)**

# findpath([path start location])