Best Path

**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**

# dont need to check for dead ends

**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**)**

# non-recursive findpath for very long paths

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

**nonlocal** bestpath

stack **=** **[(**path**,** posmoves**(**path**),** **True)]**

**while** stack**:**

newpath **=** stack**[-**1**][**0**].**copy**()**

moves **=** stack**[-**1**][**1**]**

**if** stack**[-**1**][**2**]:** # only evaluate path the first time

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

stack**.**pop**()**

**continue**

**if** isdone**(**newpath**):**

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

bestpath **=** newpath**.**copy**()**

# dont keep looking once a path is complete (optional)

# stack.pop()

# continue

stack**[-**1**][**2**]** **=** **False**

**if** **not** moves**:**

stack**.**pop**()**

**else:**

newpath**.**append**(**moves**.**pop**())**

stack**.**append**((**newpath**,** posmoves**(**newpath**),** **True))**

# findpath([path start location])