WP

Description

Pytorch is a widely used AI framework. I use it as a carrier to provide a simple and interesting game. I hope you like it.

The format of flag is flag{md5(the shortest path of the maze)}

Solution

This challenge gives us only one file, a '.pt' file. In the description I have prompted that it is a pytorch model and it contains a maze.

We can use 'torch.load' to get the model's parameter and know what's in this model by applying 'print()' function. You can find some base64 string here.

Maze(
 (maze): Linear(in_features=42, out_features=42, bias=True)
 (eW91X3Nob3VsZF9maW5kX3RoZV9zaG9ydGVzdF93YXlfZnJvbV8yX3RvXzMu):
Linear(in_features=0, out_features=0, bias=True)
 (bnVtYmVyXzBfc3RhbmRfZm9yX3dheV9hbmRfbnVtYmVyXzFfc3RhbmRfZm9yX2Jsb2Nr):
Linear(in_features=0, out_features=0, bias=True)
 (eW91X3Nob3VsZF90ZWxsX21lX3RoZV9wYXRoX2J5X3VzaW5nX0FXU0Rf):
Linear(in_features=0, out_features=0, bias=True)
)

After decoding these you can get hints:

you_should_find_the_shortest_way_from_2_to_3.
number_0_stand_for_way_and_number_1_stand_for_block
you_should_tell_me_the_path_by_using_AWSD_

You can know the maze's size is 42×42 for '(maze): Linear(in_features=42, out_features=42, bias=True)'

Let's have a look at this maze:

```
##########################################
2 # #      #      # # ##     #   #    #     #
# # # # #  # ## #      # # # #   ## ## # #
# #    # # #   # #### ##  # #  # ### # # #
# # # # ## #     ## # ## # # #    # #
# # # # # # # ####    ###   ## #   # ## #
# # ## ### #      ## # # ## # # # ##    #
# # #    # # # # # #    #   #   #   ###
# ## # # # # #  # # # # ### # # ### #
# # ## # # #   #  # # # #     # #     #
## ## #   # ## # # # # # ## #  # ## #
##    ## ## #  # # #   # # #   #   # #
# # # # # # #  ## ## # # # ### # # #
# # # # #   # # ## #  # # ##    # # ##
# # # ## ### # # #  # # # #   ### # #
```

```
 #    #    #      #  ## #  #  # #     ##  # #
# #  # ## # ## #  # #   ##  # # #### # #  #
#  ## #     ## # # #  #   #   #     #    # #
## #      ## # #  # ### # # ## ## # # # # #
 #  # # #   # # #  #    ##     ##   ### #
# ## ## # ## ##  ## # ##   # #  # # ##  ##
#         # #    #  # #  ## # #  #     # #
# ## # #  # ## ##  # #   # # ## # ### # #
# # # #  #  ###   #  # # #  # #  #   #  #
# #   # ### # ## # #   # # #  #  ## #
#  # ## #  ## ##  ## # ## ## #  # # #
## #   # # #  #  ## # #   #  # # # #
#   # # # # ## #   #  # ### ##    #  ##
## # # ## # # # ## # #   #  ## ## # #
# # # # ## # ## #   #### #   # # #
# #  # ## # # # # # ## #    # ##  # #
# # # # ## # # #  # ## # # # #  # # #
# #   #   # ## # # # #  # # ##   #
# ## # ## #   #   # # # #  # # # # ##
## #     ### # ## # # # # ## # # #  #
# # # ### # # # ## ### #  # # # #
# # # # # # # # # ## # # ### # # #
# # # #  # # # # ## # # # ##   # # #
# # # ## ## # # # #   ## # # # ###   ##
#    #      ### # # ##  ## # # # ## # #
# ## #  ## #    #     #     # #    ### 3
##########################################
```

It is eazy realizing you shoud find the shortesr way from 2 to 3. This challenge is baby if you master the Breadth-First-Search(bfs) algorithm.

You can run the 'exp.py' to get following answer:

path:
DSSSSSSSSDSSDDDWDDWDWDWDWWAWAWAWWDDSDSDDWWDDDSSSDDDDSDDDWDWDWW
DDSSSSDDSSDDWDWWWWDWDDDSSDSDDWWWDDSSSSSAASAASASSASAAWWAAASASSSASSSSA
AWAWAWWDWWWWWWAASSSSASSASAASASASSASSDSDSSAASASSAASSSSASSAAAWWAWWWDW
WAWWDWDWWWDWWAAASAAASSSAAASSSSDSSSSASSSSDSDSSSSSDDWWWWWDDDDSDSSAASS
DDDDSDDDDWWWWWWAWAAWWDWWWDWDWDWDWDWDWWAWWAWWDWDDSDSSDDSDDS
DSSASAASASSASASSASASSSSSDDSSDDSDSDDDDDWWAWWWWAWWWAWWWDDDWWDDWWWD
WWDWWDDSSDDDSSDDSSSASASSASASASSSSSDSDSDDSDSD
md5(path)----->flag{689bc7711b6becd9c1d92ae3bb9e5e59}

So the flag is flag{689bc7711b6becd9c1d92ae3bb9e5e59}