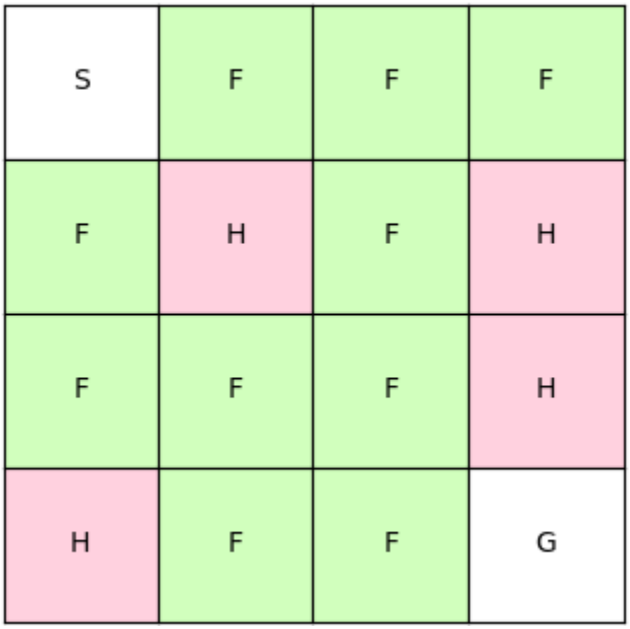
PROJECT TITLE : MONTE CARLO SEARCH TREE

PROBLEM STATEMENT

The Frozen Lake environment is a 4×4 grid which contain four possible areas Start (S), Frozen (F), Hole (H) and Goal (G). The agent moves around the grid until it reaches the goal or the hole. If it falls into the hole, it has to start from the beginning and is rewarded the value 0. The process continues until it learns from every mistake and reaches the goal eventually. Here is visual description of the Frozen Lake grid (4×4):



This project applies Monte Carlo Tree Search (MCTS) to a deterministic variant of the FrozenLake environment (4x4). We use the Upper Confidence Bounds for Trees (UCT) algorithm. UCT is the most popular MCTS algorithm.

The agent can take 4 possible actions:

0 left
1 down
2 right
3 up

DATASET

FrozenLake-v0
The agent controls the movement of a character in a grid world. Some tiles of the grid are walkable, and others lead to the agent falling into the water. Additionally, the movement direction of the agent is uncertain and only partially depends on the chosen direction. The agent is rewarded for finding a walkable path to a goal tile.

Reference links of our Dataset :
https://gym.openai.com/envs/FrozenLake-v0/
https://github.com/openai/gym/blob/master/gym/envs/toy_text/frozen_lake.py

RESEARCH PAPER

https://castlelab.princeton.edu/html/ORF544/Readings/MCTS%20tutorial%202012.pdf
The copy of the paper attached in the mail .

INFERENCE

MCTS is better for large state space problems and many complex games as it does not stick to the current optimal path, instead it continuously keeps checking for other better optimal paths while holding the exploration- exploitation trade off at all times.