

Practical 2 Exercise

Part I: Blind Search

1. Modify the Step 3 in Section B to define a function named **find_shortest_path()** to return the shortest path amongst all. For example from node D to node F, it should return ['D', 'B', 'E', 'F']. (Hint: you may check the size of each list and find the smallest one)
2. Modify the Depth-first search algorithm in Section C to define a function named **depth_limited_search()** to return a path within the given depth-limit.
3. Two families are trapped on a sinking ship in an ocean (which may take few hours to sink completely). Each family consists of a father, a mother and a child. They discover a small island not very far away, but all of them do not know how to swim (and it is really freezing cold to swim over to the island). There is only 1 small raft, which can occupy at most 2 people at a time, regardless the weight or size of the persons. They agree to sail back and forth to help each other, but with the following conditions:
 - Only adults (father or mother) can operate the boat.
 - The children are too afraid to be left alone without any of their parents, even with the presence of the parent(s) from another family. So one of the parents must stay together with his/her own kid.

Assume that it may take 10 minutes for a single trip to travel from the ship to the island using the raft (and vice versa, assume that no wind and waves that can bring them to somewhere else), and the ship is estimated to sink completely 150 minutes later.

Apply **breadth-first search** and **depth-limited search** separately to return a possible path, so that all members from the two families can reach the island safely before the ship sinks into the ocean totally.

Part II: Heuristic Search

1. Add the **simpleHillClimbing** function to the code in **EightPuzzle.py** to get a path that reaches the goal state from the start state. Then call and run the **simpleHillClimbing** function in **heuristic_search.py**. Check whether the algorithm could guarantee completeness and optimality.

Reference:

1. <https://bastam.trinket.io/first-day-get-pythonic#/everything-is-a-dictionary/what-is-it-an-english-dictionary>
2. <https://www.python.org/doc/essays/graphs/>
3. <http://eddmann.com/posts/depth-first-search-and-breadth-first-search-in-python/>
4. https://github.com/marianafranco/missionaries-and-cannibals/blob/master/python/missionaries_and_cannibals.py