Task 1

Goal: Implement the general search algorithm. Experiment with Best-First and A* search on the 8-puzzle, test additional heuristics that I provide. Test these additional heuristics for the Num-berlink puzzle with A* search. Compare the resulting search for the different heuristics.

For the 8-puzzle, use the goal state here:

1 2 3

8 _ 4

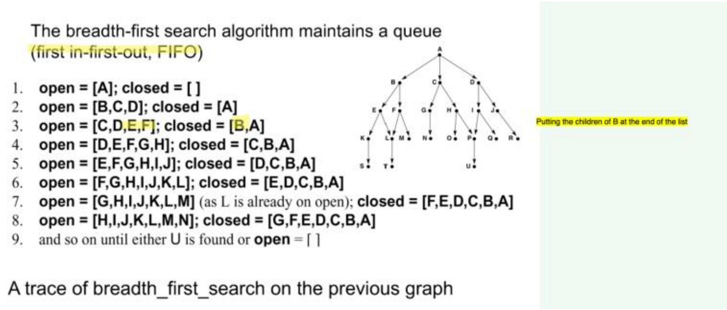
7 6 5

Input: a state is a board configuration in form of a list. The board configuration of the goal state could be represented as the string: (1 2 3 8 B 4 7 6 5). Remember, that not all initial configurations can be transformed into this goal configuration, take care with your test examples to be solvable.

Output: A list of successive states, beginning with the start state and ending with the goal state, and including all states required to transform the start state into the goal state.

Description:

1. Implement a general search (based on the pic below), with OPEN and CLOSED lists. Implement different OPEN list orderings (for breadth- first, depth-first, Best-First, and A*)



The breadth-first search algorithm maintains a queue (first in-first-out, FIFO)

1. open = [A]; closed = [ ]
2. open = [B,C,D]; closed = [A]
3. open = [C,D,E,F]; closed = [B,A]
4. open = [D,E,F,G,H]; closed = [C,B,A]
5. open = [E,F,G,H,I,J]; closed = [D,C,B,A]
6. open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]
7. open = [G,H,I,J,K,L,M] (as L is already on open); closed = [F,E,D,C,B,A]
8. open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]
9. and so on until either U is found or open = [ ]

A trace of breadth_first_search on the previous graph

Putting the children of B at the end of the list

2. Implement a successor state generator for the 8-puzzle

3. Implement Hamming distance, Manhatten distance, Permutation Inversion heuristics for the 8-puzzle

4.Compare the length of the search path and the optimality of the solution found for Best-First and A* for the three admissible heuristics mentioned above.(Also, implement this for linear conflict and hamming distance)

Task 2

Numberlink is a type of logic puzzle where the goal is to find paths to connect numbers in a grid.

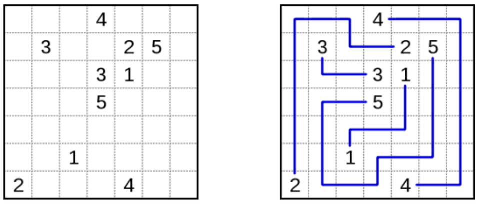


Figure 2: Numberlink Puzzle problem and solution

To solve the Numberlink puzzle you need to make sure that you are not just placing a set of paths into a grid to get an answer. You need to ensure that there is only one solution, your solution. The rules to link the numbers in the grid are:

- Each pair of numbers needs to be connected with a continuous line that only passes vertically or horizontally between squares.
- Lines cannot pass through the same square twice.
- No two lines can cross each other or occupy the same square.

Numberlink puzzle

1. You have to Implement the following state representation scheme for the search algorithms developed here. Don't just do it for a 4x4 Numberlink puzzle. Make it generalized.

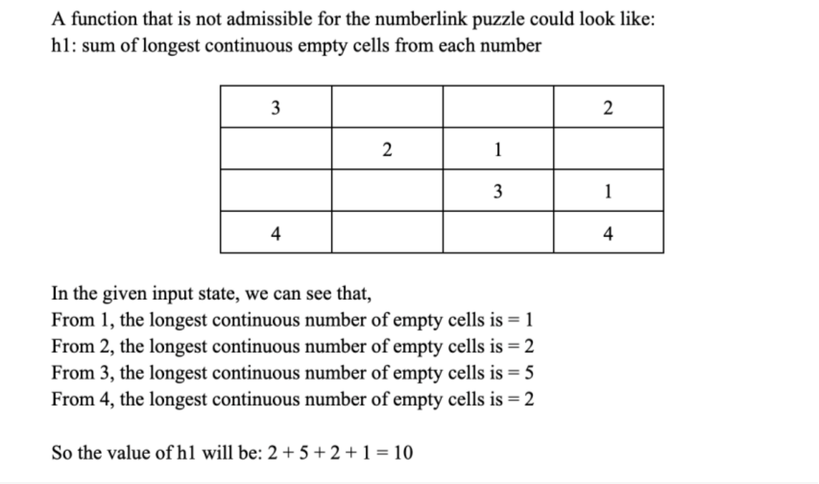
[The following is a representation scheme for a 4x4 numberlink puzzle using only parentheses, alphanumeric characters, punctuation and [possibly nested] list structures:

[[#, #, #, #], [#, *, #, #], [#, *, *, #], [#, #, #, *]]

A numberlink puzzle is a square dimension, So, it will have ‘n’ columns and ‘n’ rows. In this representation scheme, the numbers are represented using ‘#’ and the blank spaces using ‘*’. Here, a nested list structure is being used to show that the numberlink puzzle has same number or rows and columns.]

2. Develop a successor state generator for the Numberlink puzzle

3. Compare the length of the search path and the optimality of the solution found for Best-First and A* for Manhattan Distance and for the inadmissible heuristics (Pic on the right)



A function that is not admissible for the numberlink puzzle could look like:
h1: sum of longest continuous empty cells from each number

3			2
	2	1	
		3	1
4			4

In the given input state, we can see that,
From 1, the longest continuous number of empty cells is = 1
From 2, the longest continuous number of empty cells is = 2
From 3, the longest continuous number of empty cells is = 5
From 4, the longest continuous number of empty cells is = 2

So the value of h1 will be: 2 + 5 + 2 + 1 = 10