Bilkent University

CS 461
Artificial Intelligence
Homework 2
Report

Güneş Batmaz
Alper Kağan Kayalı
İrem Saka
Mert Sargon
Arif Can Terzioğlu

Instructor: Varol Akman

Brief Explanation

Our program aims to solve the The Eight Puzzle using Beam Search. To do that, firstly, we implemented a Puzzle Generator that generate randomized, solvable n many(for this case, n = 10) distinct initial states and writes that into a .txt file. We used the .txt file to be able to use the same distinct initial states for comparison. Then, we implemented a Puzzle Solver that solves the generated puzzles with Beam Search that implemented inside that class. Also, in the same class, we use Graph Stream library to visualize the solution, and Apache Poi to write the number of the length and closed nodes to generate the required plots.

Platform Used: NetBeans IDE 8.2
Outside Source : Our UI was built with Graph Stream library and Apache Poi

- The listing of the 10 distinct states.

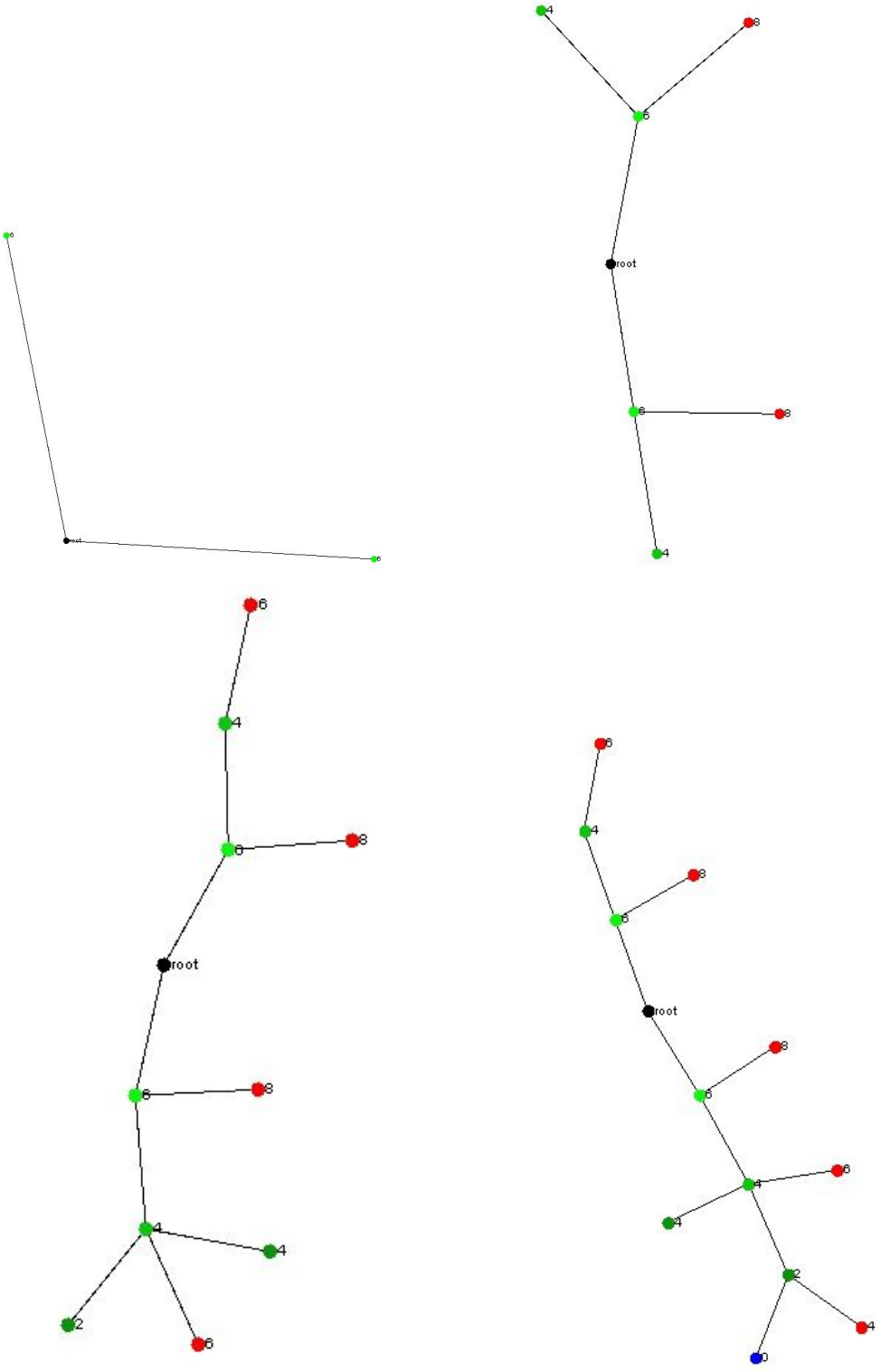
```
1 3 0
4 2 5
7 8 6
-------
0 1 3
4 2 6
7 5 8
------
1 2 3
4 0 5
7 8 6
------
1 2 0
4 6 3
7 5 8
------
1 0 3
4 2 6
7 5 8
```

```
1 2 3
4 6 0
7 5 8
------
1 2 0
4 5 3
7 8 6
-------
1 2 3
7 4 5
0 8 6
------
1 2 3
4 5 6
0 7 8
------
1 0 3
4 2 5
7 8 6
```

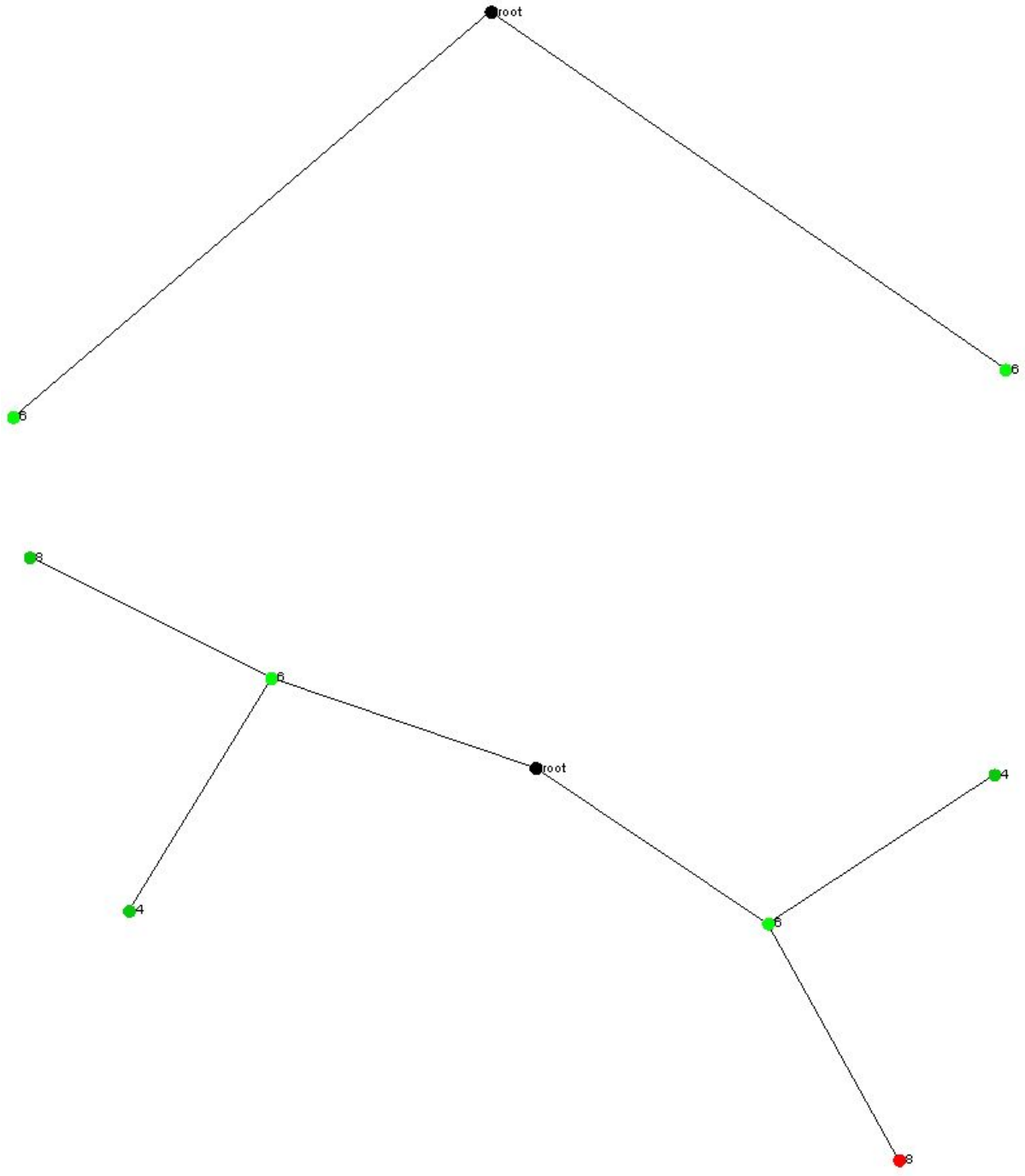
And the first states is S, that is used in following parts of the report, i.e.

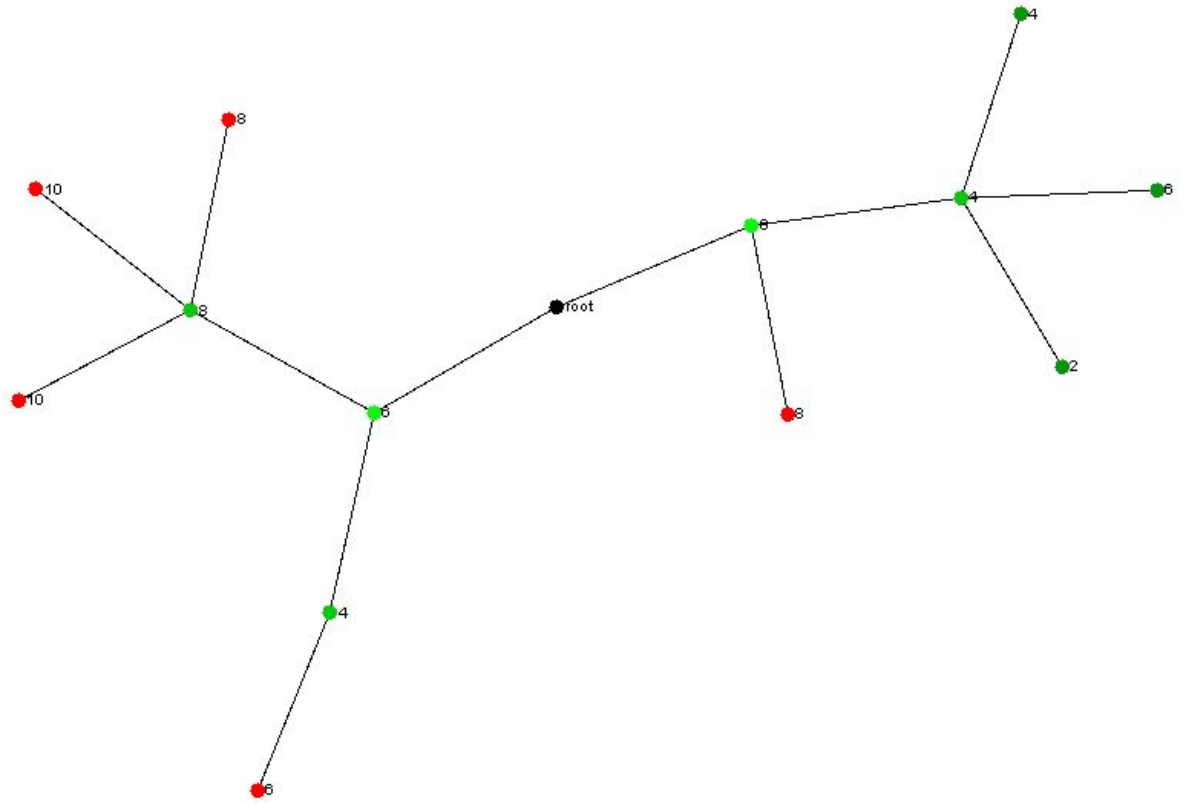
```
     1 3 0
S =  4 2 5
     7 8 6
```

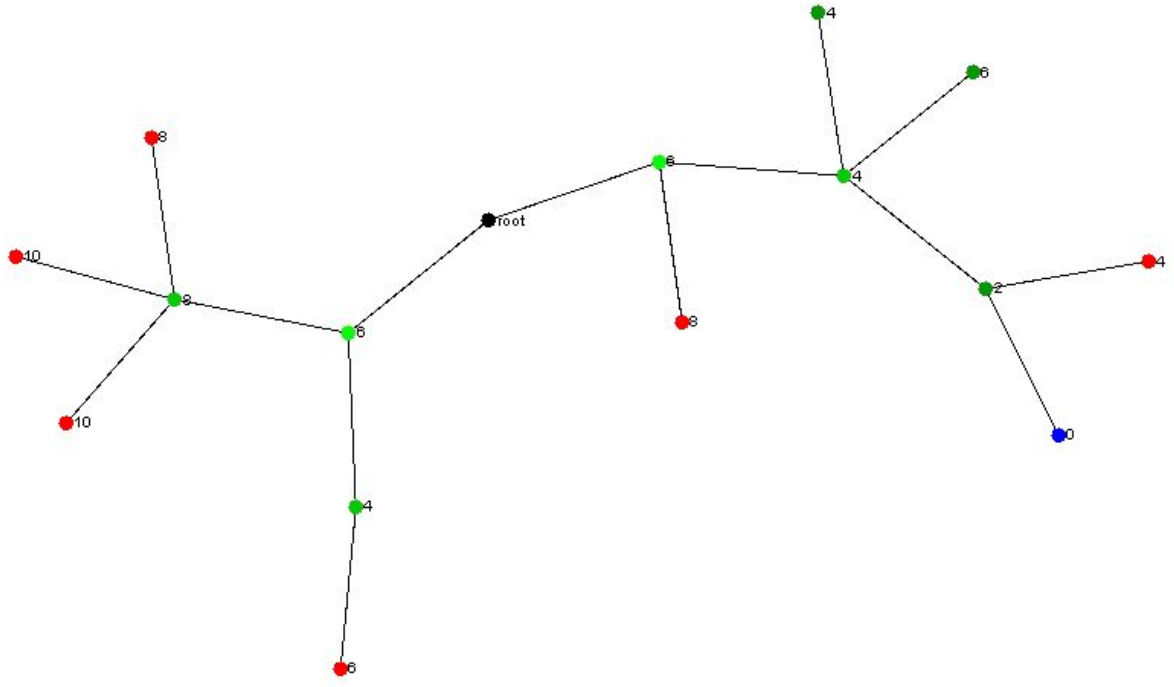
Sample graphical solution for S (defined above), and using beam width w = 2. It is shown step by step and every node is coloured according to its level. You may see the coloured version from the soft copy submission.



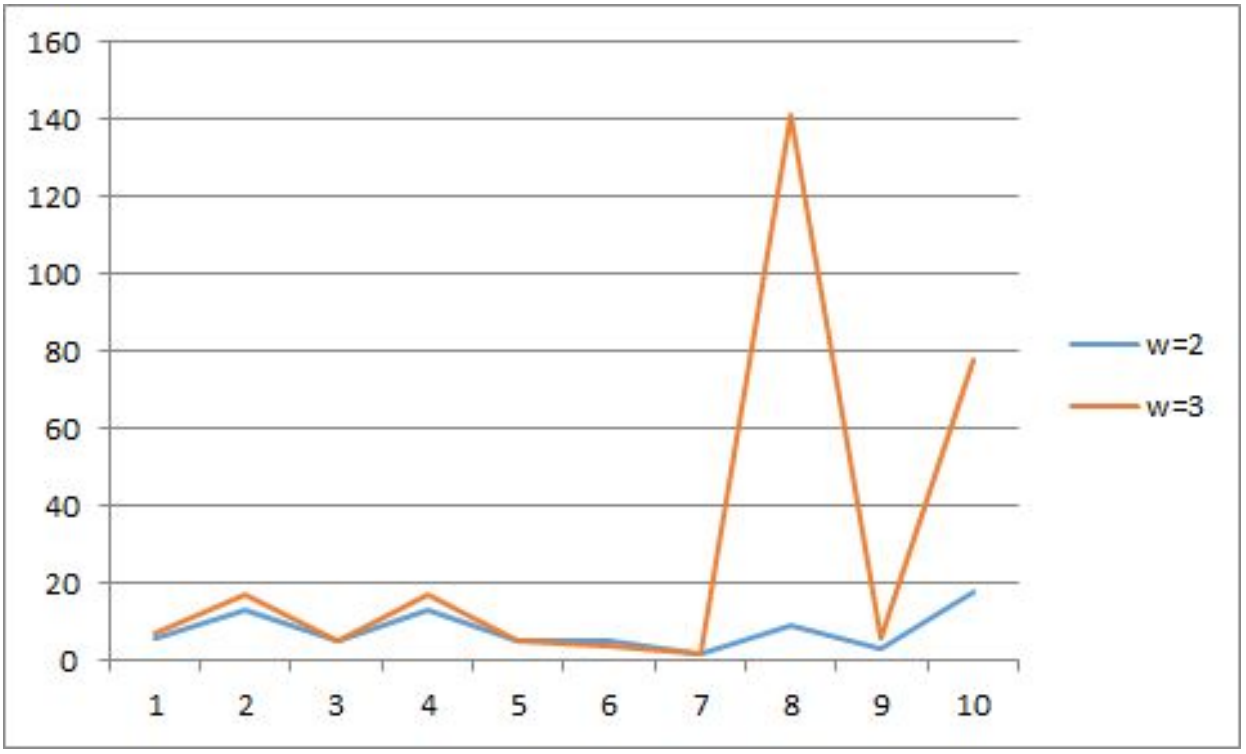
Sample graphical solution for the same S and using w = 3. fully tracing the individual moves of program to reach the goal is as follows.




4
8
10
6
8
8
foot
10
2
6
8
4
6


4
6
8
6
4
foot
10
8
4
8
10
2
0
4
6

Number of Closed Nodes in the solution of the each initial state has been demonstrated as below:

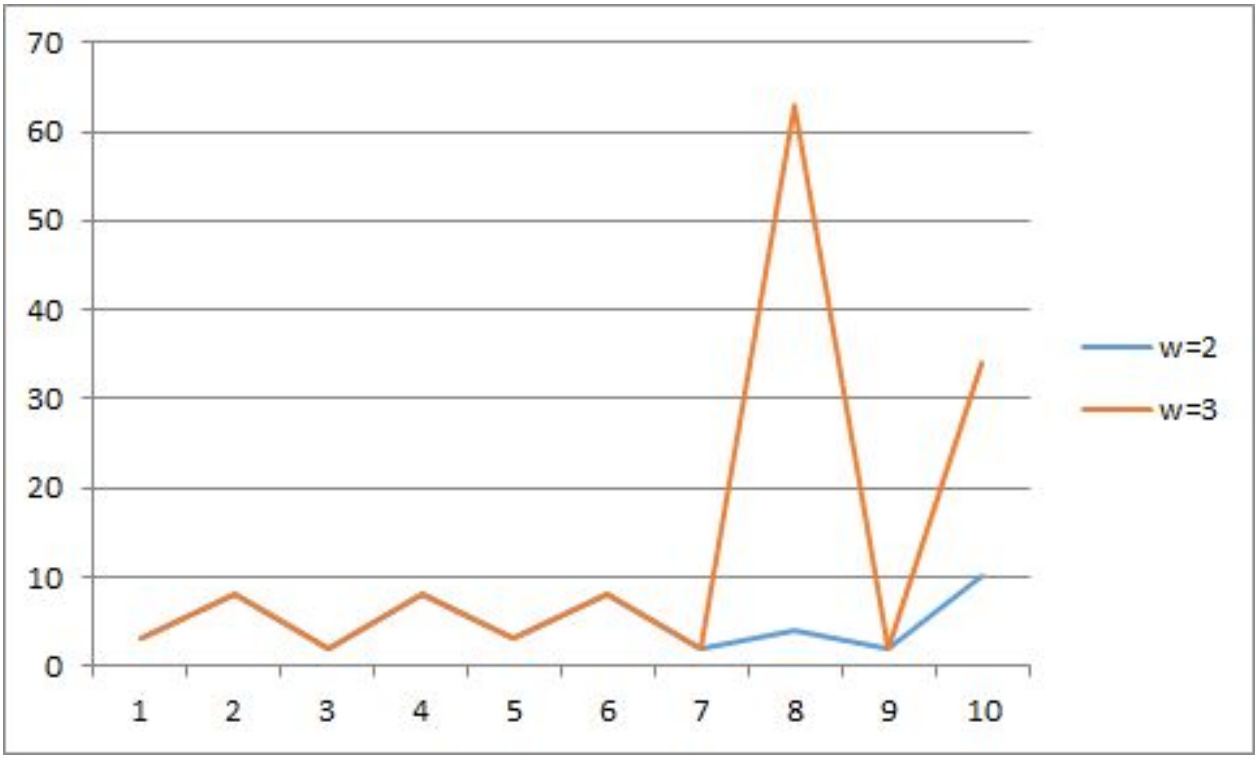


And the values can be seen distinctly from the following table:

	Closed Nodes	
ID	w = 2	w = 3
0	6	7
1	13	17
2	5	5
3	13	17
4	5	5
5	5	4
6	2	2
7	9	141
8	3	6
9	18	78

As you may see from the graphical display above, the number of closed nodes when w = 3 is almost always higher that w = 2. It satisfy the expectations as there are more nodes that are evaluated in w = 3 then w = 2.

The length of the solution for each initial state has been demonstrated as below:



And the values can be seen distinctly from the following table:

	Solution Length	
ID	w = 2	w = 3
0	3	3
1	8	8
2	2	2
3	8	8
4	3	3
5	3	3
6	2	2
7	4	63
8	2	2
9	10	34

Normally in the long run, when w = 3, the time that the goal state found should be smaller than the w = 2 since it evaluates more nodes, but because of our initial states-as they are simple- these graphics are obtained.