Assignment #2

Artificial Intelligence - CSCE 523

Due: 8:00 AM, Wednesday February 6, 2019

Search and Game Tree Search

Turnin: E-mail me a zip file containing your typed solution to questions 1 and 2,

and your program and report for question 3.

1. (30 points) For the following maze, show the lists of open and visited nodes (with their associated costs) for each cycle of the listed search algorithms. The start cell is S, and the goal cell is G. The agent can move north, south, east, and west. The agent expends 1 point moving south or west, and 2 points moving north or east.
   1. Decide on a heuristic estimator function and write the function out.

**Sol’n:** Heuristic estimate function will be Manhattan distance

Let goal node , and an arbitrary node , then the Manhattan Distance, h, between them is given by the following equation

* 1. Decide on a method to break ties (label all cells with letters and break alphabetically, first come first served, etc.) and write that out

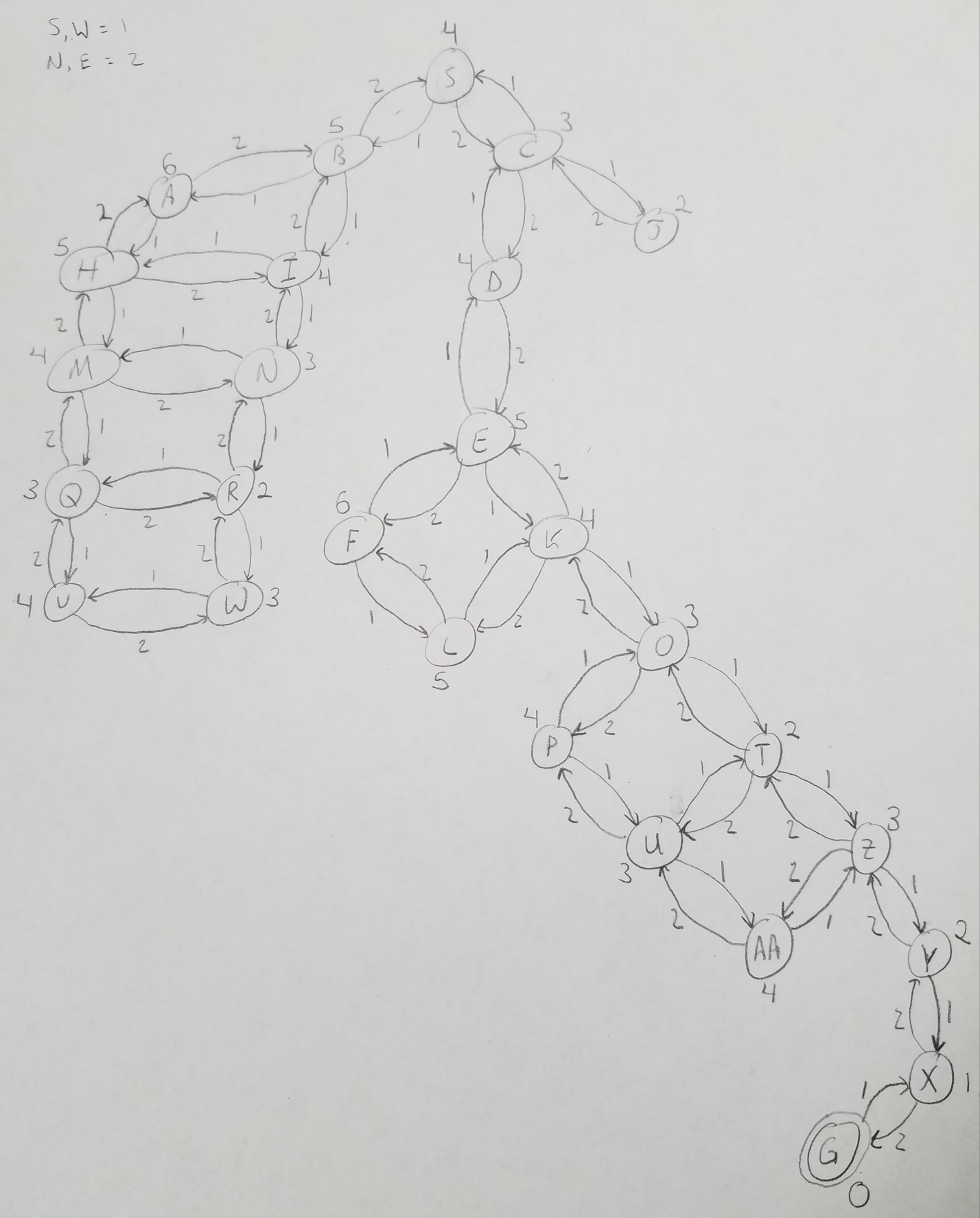
**Sol’n:** Tiebreaker will be alphabetical using the labels shown below.

Then perform the following searches on the space (follow the format from

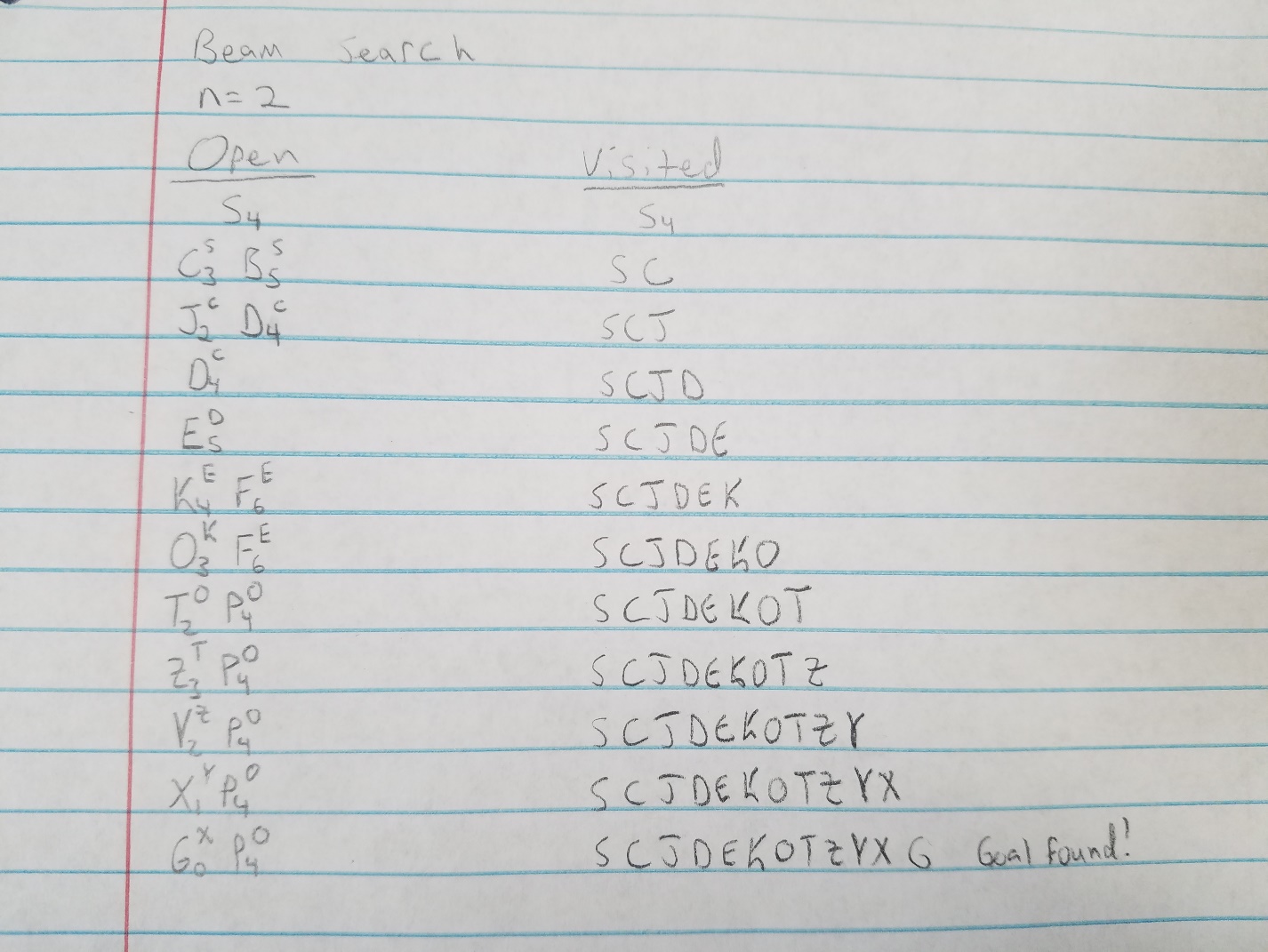
class):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **S** | **C** | **D** | **E** | **F** |
| **H** | **I** |  | **J** |  | **K** | **L** |
| **M** | **N** |  |  |  | **O** | **P** |
| **Q** | **R** |  | **G** |  | **T** | **U** |
| **V** | **W** |  | **X** | **Y** | **Z** | **AA** |

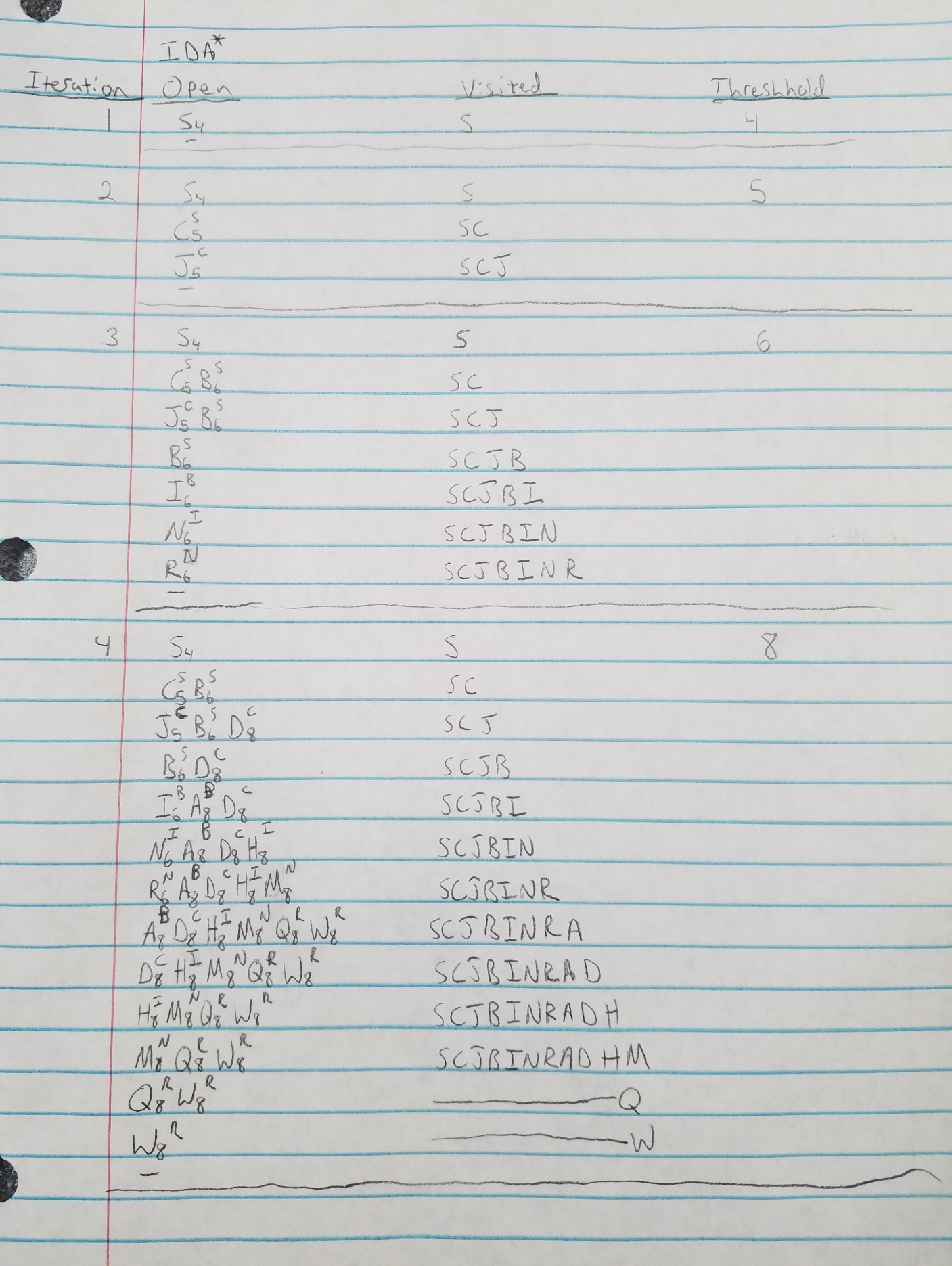
Resulting Graph with heuristic values

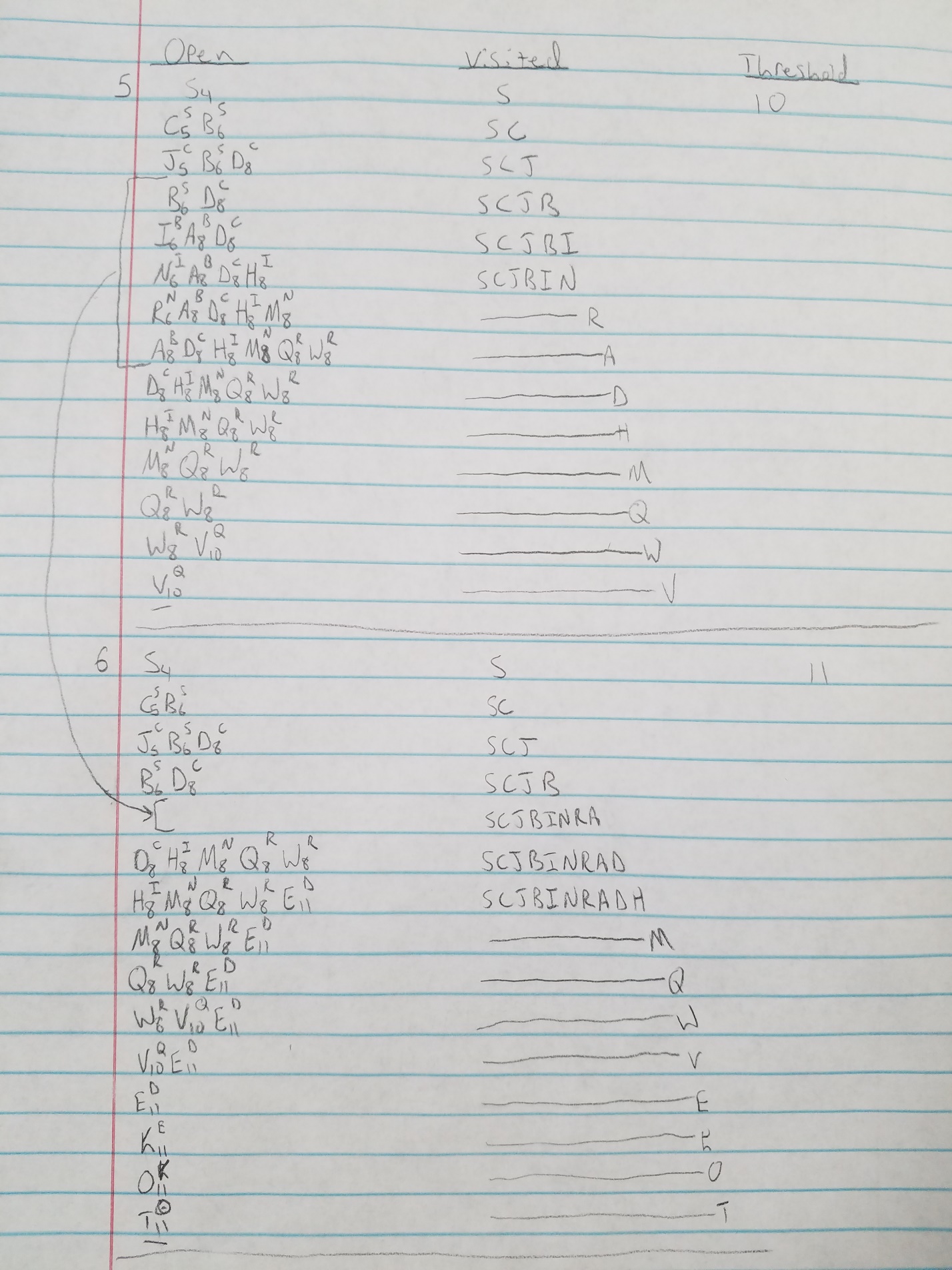


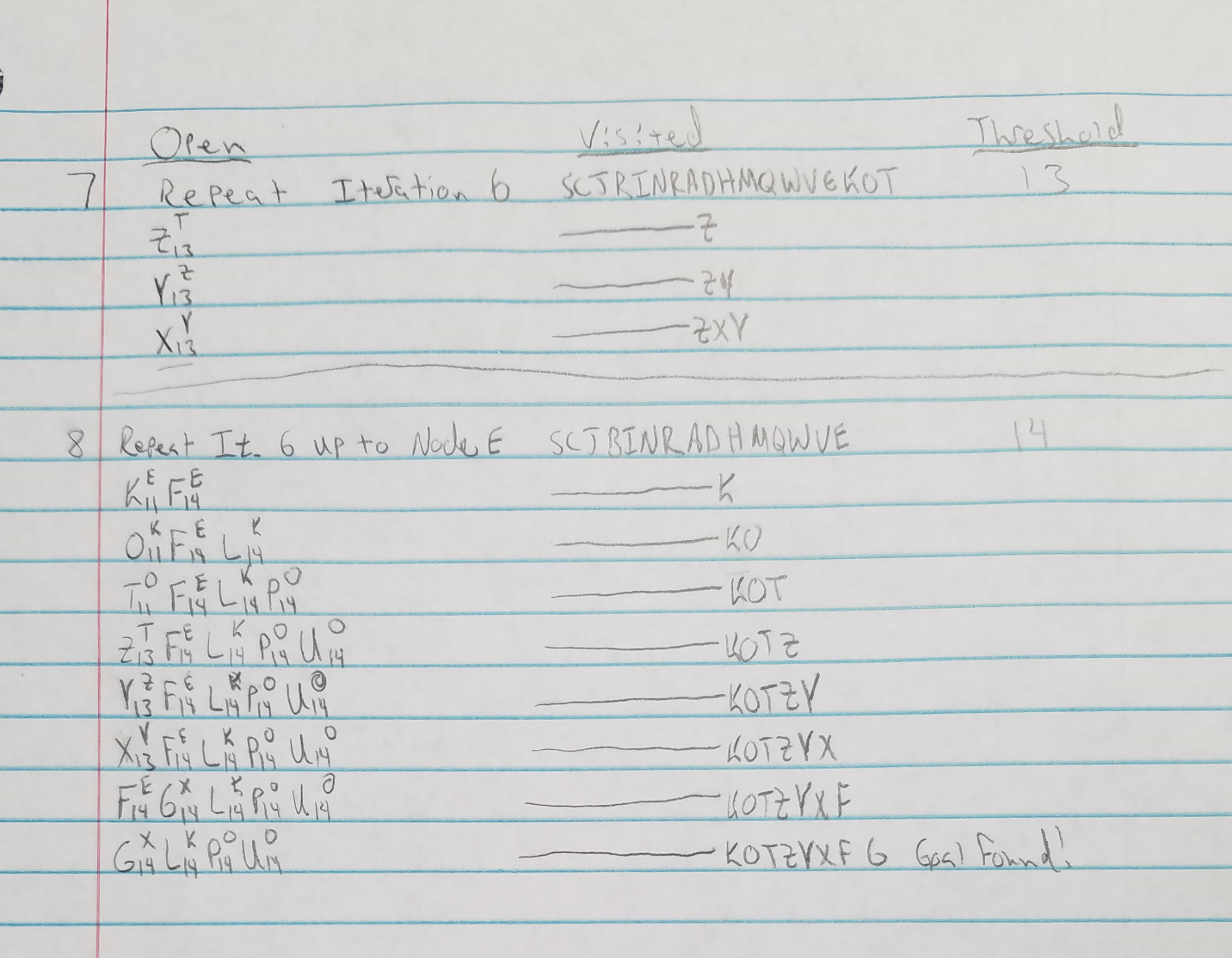
* 1. Beam search with a beam size of 2



* 1. IDA\* search







Note: If I had it to do over again, would have used a different heuristic or ordering scheme

1. (20 points) For the following Light-Up puzzle, show the sequence of variable assignments during backtracking with forward checking; examine cells in alphabetical order. Show assignments by writing the forward checking table process (14 columns: step number, a value, b value, …, l value, backtrack {list the constraint violation that causes the backtrack}).

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |
| e | **1** | **2** | f |
| g |  | **0** | h |
| i | j | k | l |

**Sol’n:**

**Constraints:**

* Must be a light in c or f (based on black 2 block)
* Must be a light in either b or e (based on black 1 block)
* No light in one of k or h (based on black 0 block)

Notes:

* v indicates a light. An x indicates no light can be placed in the cell (forward checking)
* Green fill indicates a valid variable assignment (one that causes no constraint violations)
* Red fill indicates a variable assignment that causes a constraint violation
* Move ordering done alphabetically (per instructions)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Step** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **Constraint violation (backtrack)** |
| 1 | v | x | x | x | x |  | x |  | x |  |  |  | Must be a light in b or e & c, caused by putting light in a |
| 2 | x | v | x | x |  |  |  |  |  |  |  |  | Must be a light in c, caused by putting light in b |
| 3 | x | x | v | x |  |  |  |  |  |  |  |  | None |
| 4 | x | x | v | x | v |  | x |  | x |  |  |  | None |
| 5 | x | x | v | x | v | v | x | x | x |  |  | x | None |
| 6 | x | x | v | x | v | v | x | x | x | v | x | x | None. Puzzle Solved |

1. See the attached writeup and code for the solution to problem 3