

# CSC 242: Homework 4

Assigned on Mar. 17  
Hand in on Mar. 31 (Tuesday) in Class

## Requirement

Please hand in the hard copy of your homework in our class (that is, before 2pm) on on Mar. 31 (Tuesday). If you need to use your delay coupon, you can hand in your homework to my office. Include the **class ID number** assigned to you for this class, **your name**, and **how many days** late it is (if handed in late) in the headline. The ID number is used to simply the grading procedure. If you do not know your ID number assigned to you, you can check the link.<sup>1</sup> If you are not on the list, please contact our TA Sean Esterkin (sesterki@u.rochester.edu). (\*), (\*\*), or (\*\*\*) indicates the difficulty of each question.

## Policy

We apply the late policy explained in syllabus to all homework. Any grading questions must be raised with the TA in two weeks after the homework is returned.

The homework must be completed individually. However, you are encouraged to discuss the general algorithms and ideas with classmates in order to help you answer the questions. You are also allowed to share examples that are not on the homework in order to demonstrate how to solve problems. If you work with one or more other people on the general discussion of the assignment questions, please record their names over every question they participated.

However, the following behaviors will receive heavy penalties (lose all points and apply the honest policy explained in syllabus)

- explicitly tell somebody else the answers;
- explicitly copy answers or code fragments from anyone or anywhere;
- allow your answers to be copied;
- get code from Web.

---

<sup>1</sup>[https://docs.google.com/spreadsheets/d/1hgxcxqFxBxZYb55\\_HHi2pWQt0rEyEppcStSZBKni8MU/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1hgxcxqFxBxZYb55_HHi2pWQt0rEyEppcStSZBKni8MU/edit?usp=sharing)

## 1 (\*) Terminologies: 2 points

Define in your own words the following terms: state, state space, successor function, initial state set and goal state set.

## 2 (\*\*) Rubic's Cube: 4 points

Consider the task of solving Rubic's Cube. Please formulate it as a state-space search problem. Specifically, please define the state space and the successor function. For simplicity, let us consider a  $2 \times 2 \times 2$  cube (Let me be a nice guy).

## 3 (\*\*) Complexity: 2 points

Consider the performance of search algorithms on trees. Recall the complexity analysis for Bidirectional search with BFS. Its time complexity is  $O(b^{\frac{d}{2}})$  and its space complexity is  $O(b^{\frac{d}{2}})$  where  $b$  is the branching factor (assume finite) and  $d$  is the goal depth. If we apply the Bidirectional search with iterative deepening, what would be the space complexity and time complexity?

## 4 (\*\*) Search Algorithms: 10 points

Little Red Riding Hood wants to go to her grandmother's house in the forest to give her some cake. The forest is a dangerous place where the Big Bad Wolf resides. Little Red Riding Hood's mother has instructed her not to venture into the forest and to only go along the road. The following  $5 \times 5$  grid shows the area that Little Red Riding Hood needs to traverse. She starts from square a (denoted by R). Her grandmother's house is located at square s (denoted by G). The forests are denoted by F (on squares f, i, m, r, u and y). Your task is to help Little Red Riding Hood get to her grandmother's house. Assume that she can only move in four directions namely left, down, right and up. She cannot travel diagonally and will not go into the forests. Also assume that the successor function will cause legal moves to be examined in an counter-clockwise order: left; down; right; up. Note that not all of these moves may be possible for a given square (squares with forest on them will never be examined).

a	b	c	d	e
R				
f	g	h	i	j
F			F	
k	l	m	n	o
		F		
p	q	r	s	t
		F	G	
u	v	w	x	y
F				F

1. Using Depth First Search, list the squares in the order they are expanded (including the goal node if it is found). Square “a” is expanded first (hint: Square “b” will be examined next). Assume **cycle checking is done** so that a node is not generated in the search tree if the grid position already occurs on the path from this node back to the root node (i.e., Path checking DFS). Write down the list of states you expanded in the order they are **expanded**. Write down the solution path found (if any), or explain why no solution is found.
2. Using Breadth First Search, write down the list of states you expanded in the order they are expanded (until the goal node is reached). Use the same cycle checking as in the previous question.
3. Using Iterative Deepening Search, draw the trees built at each depth until a solution is reached. Use the same cycle checking as in the previous questions.
4. Consider the heuristic function  $h(\cdot)$  which is the Manhattan distance<sup>2</sup> between a given square and the goal square. Is  $h(\cdot)$  an admissible heuristic?
5. Perform  $A^*$  search using the heuristic function  $h(\cdot)$  defined in the previous question. In the case of ties, expand states in alphabetical order. List each square in the order they are added to the OPEN list, and mark it a state with a number indicating when it was expanded (Square “a” should be marked 1). Highlight the solution path found (if any), or explain why no solution is found.

---

<sup>2</sup>[http://en.wiktionary.org/wiki/Manhattan\\_distance](http://en.wiktionary.org/wiki/Manhattan_distance)