Homework 01

Submission Notices:

- *Conduct your homework by filling answers into the placeholders given in this file (in Microsoft Word format). Questions are shown in black color, instructions/hints are shown in italic and blue color, and your content should use any color that is different from those.*
- *After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,*
  <StudentID-1>_<StudentID-2>_HW01.pdf (Student IDs are sorted in ascending order)
  *E.g., **2252001_2252002_HW01.pdf***
  and then submit the file to Moodle directly WITHOUT any kinds of compression (.zip, .rar, .tar, etc.).
- *Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.*
  1. *Wrong file/filename format, e.g., not a pdf file, use "-" instead of "_" for separators, etc.*
  2. *Disorder format of problems and answers*
  3. *Conducted not in English*
  4. *Cheating, i.e., copy other students' works or let the other student(s) copy your work.*

Problem 1 (1pt) Choose True or False for each statement to indicate whether the statement is true or false and then give your explanation.

Score	Statement	True/False	Explanation
0.25pt	Iterative deepening search involves re-running breadth-first search repeatedly with increasing depth limits	False	Iterative deepening search involves re-running depth-limited search repeatedly with increasing depth limits until a goal found or a failure value is returned.
0.25pt	Tabu search behaves the same as graph search because they both attempt to avoid revisiting previously seen states.	False	Tabu search allow to revisit the old state if it was pop out from tabu list, but graph search never revisit the old state.
0.25pt	An optimal solution path for a search problem with only positive costs will never have repeated states.	True	Assume that the optimal solution path includes a cycle, so always has a solution which is better than that solution because if we reject the cycle, we will get a more optimal solution.
0.25pt	If h1 and h2 are admissible search heuristics, then h3 = 2*h1 - h2 must also be admissible.	False	Assume that we have h*(n) = 5, h1(n) = 4.5, h2(n) = 1 at state n. So we have h3(n) = 2 * 4.5 – 1 = 8 > h*(n) = 5. So in this case, h3 is not admissible.

Problem 2 (2pt) You are given the initial state (i) and the goal state (ii) of an 8-puzzle as shown below.

2	8	3
1	6	4

(i)

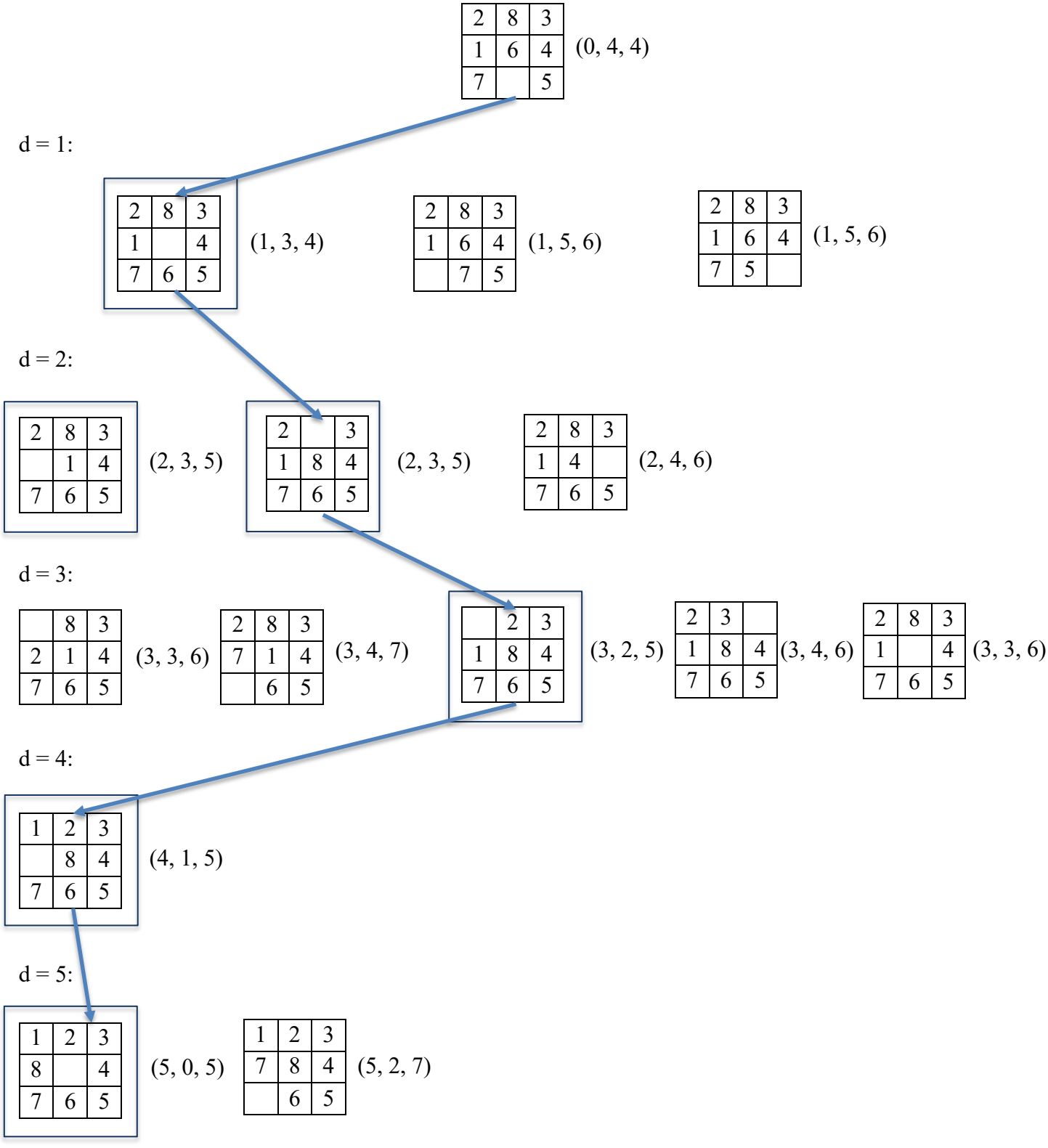
1	2	3
8		4

(ii)

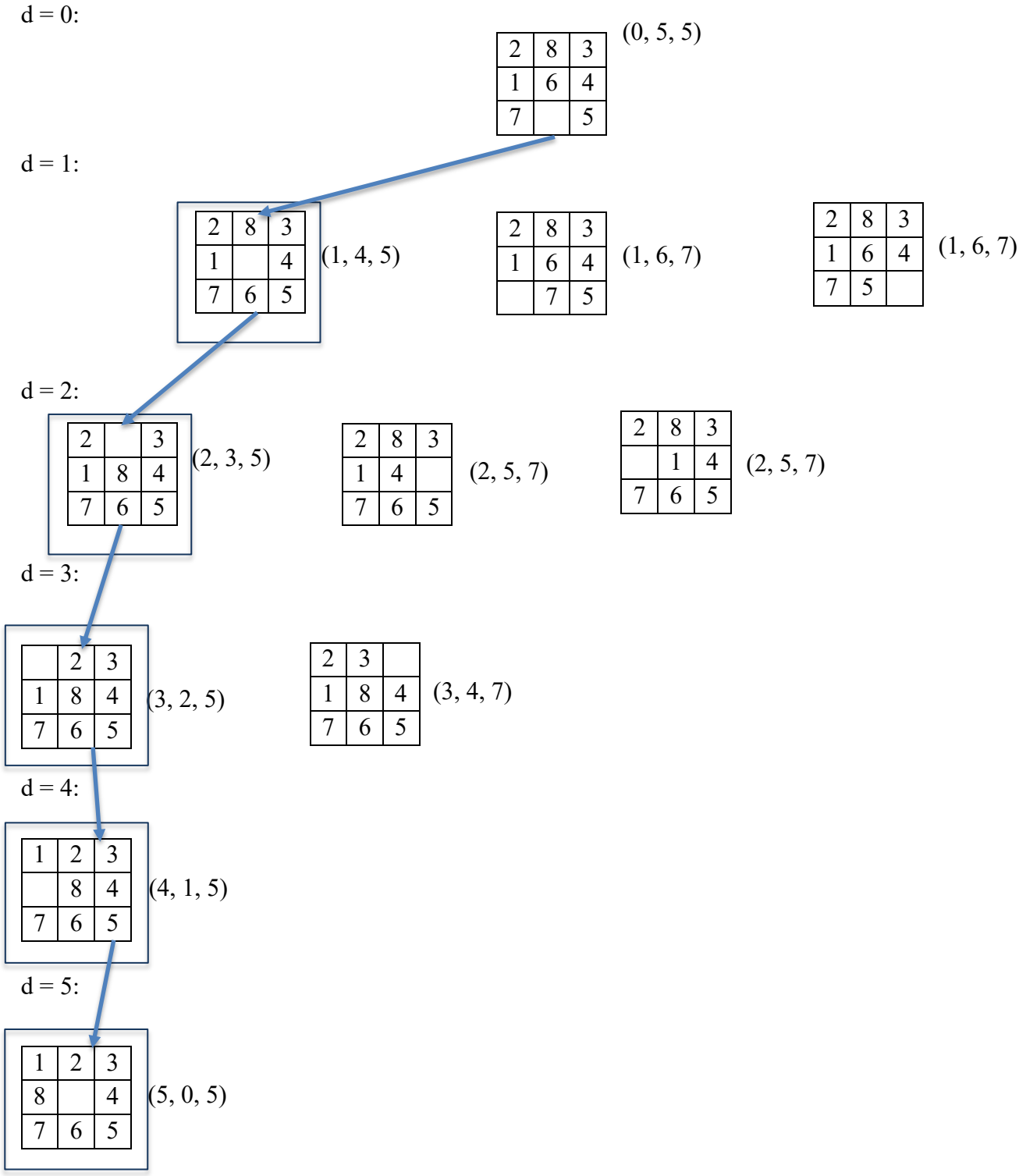
7		5

7	6	5

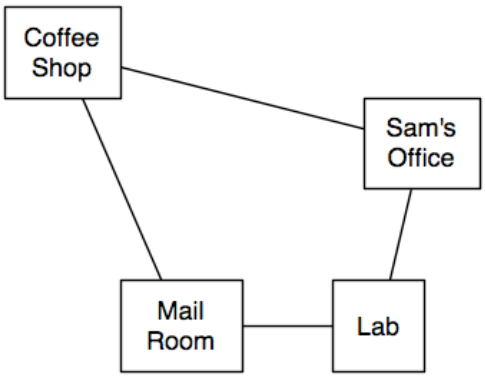
a) (1pt) Apply graph-search A* with heuristic **h_1 = the number of misplaced numbered tiles**. Ties are broken randomly.
   - Draw the search tree that includes all states generated during the algorithm procedure.
   - Compute the triple (g, h, f) for each state.
   - Mark the optimal strategy found.



b) (1pt) Repeat the question a) with heuristic $\mathbf{h_2}$ = **the sum of the (Manhattan) distance of every numbered tile to its goal position.**



Problem 3 (2.5pts) Consider a delivery robot world with mail and coffee to deliver.
Assume a simplified domain with *four locations* as shown aside. This domain is quite simple, yet it is rich enough to demonstrate many of the problems in representing actions and in planning.

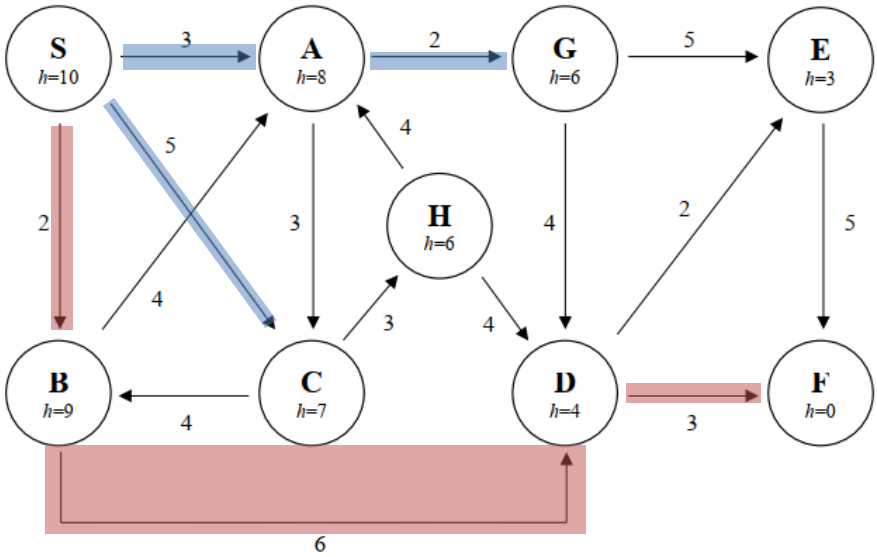


The robot, called Rob, can *pick up coffee at the coffee shop, pick up mail in the mail room, move, and deliver coffee and/or mail*. Delivering the coffee to Sam's office will stop *Sam from wanting coffee*. There can be *mail waiting at the mail room* to be delivered to Sam's office.
Rob can *move clockwise (mc)* or *move counterclockwise (mcc)*. Rob can *pick up coffee (puc)* if Rob is at the coffee shop and it is not already holding coffee. Rob can *deliver coffee (dc)* if Rob is carrying coffee and is at Sam's office. Rob can *pick up mail (pum)* if Rob is at the mail room and there is mail waiting there. Rob can *deliver mail (dm)* if Rob is carrying mail and he is at Sam's office. Assume that it is only possible for Rob to do one action at a time.
Formulate the task above as a search problem by determining the primary concepts.

Please write your answer to the following table.

Score	Search concepts	Descriptions
0.5pt	Representation for a state	The state should represent everything relevant about Rob and environment that can affect action and the goal. Each state can be described by a tuple: (location, has_coffee, has_mail, mail_waiting, sam_wants_coffee)
0.5pt	State-space graph: how many states there are and how they connect together	There are 4 (locations) * 2 (has_coffee) * 2 (has_mail) * 2 (mail_waiting) * 2 (sam_wants_coffee) = 64 states. They connect together by all of the actions. In this situation, Rob has 2 actions which always possible for every state (mc and mcc) and 4 remaining actions are conditional actions. So each state can has 2-6 edge to its childs.
0.5pt	Set of actions	mc, mcc, puc, dc, pum, dm
0.5pt	(Transition model	mc, mcc: change location based on clockwise of counterclockwise. puc: if location is Coffee Shop and has_coffee is false, set has_coffee to true. dc: if location is Sam's Office, has_coffee is true and sam_wants_coffee is true, set has_coffee and sam_wants_coffee to false.
0.5pt	Action cost	Each action has cost 1.

Problem 4 (2.5pts) Consider the following graph. The start and goal states are **S** and **F** respectively.

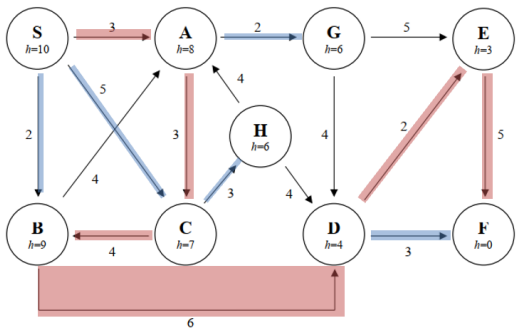
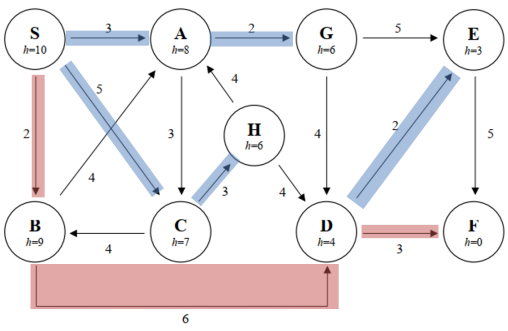


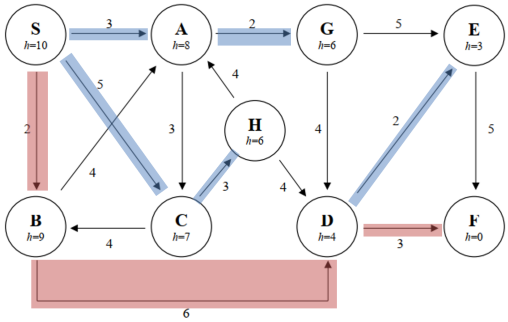
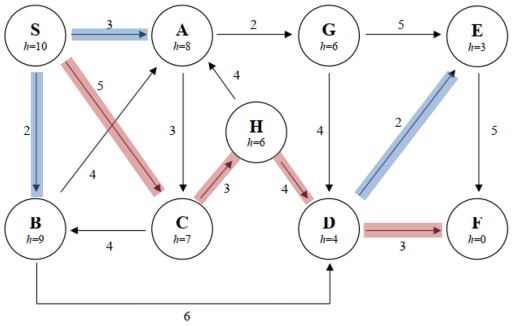
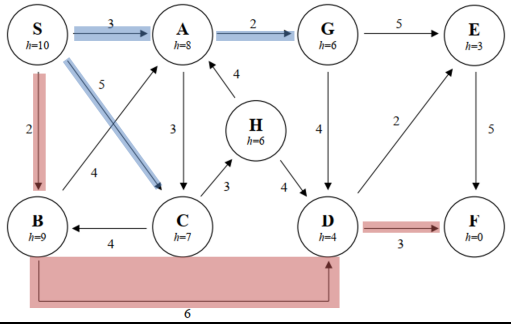
For each of the following strategies, work out order in which states are expanded, as well as the path and show the search tree used to find this solution. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first.

a) Tree-search depth-first search (DFS)
b) Breadth-first search (BFS)
c) Uniform cost search (UCS)
d) Graph-search greedy best first search (GBFS) with the heuristic h shown on the graph
e) Graph-search A* with the same heuristic.

Note that

- Tree-search DFS avoids repeated states by checking new states against those on the path from the root to the current node.
- The early goal test is applied in DFS, BFS, and GBFS.

Score	Algorithm	List of expanded states (in exact order)	Path Returned	Search tree
0.5pt	DFS	*S A C B D E*	*S A C B D E F*	
0.5pt	BFS	*S A B C D*	*S B D F*	

0.5pt	UCS	*S B A C G D H E F*	*S B D F*	
0.5pt	GBFS	*S C H D*	*S C H D F*	
0.5pt	A*	*S B A G C D F*		

Problem 5 (2pts) Present your study about Recursive best first search (RBFS).

a) (0.5pt) State the principle of the Recursive Best-First Search (RBFS) algorithm.
RBFS explores the most promising paths first, guided by an evaluation function f(n) = g(n) + h(n) like A. But it does so recursively, and only remembers the best alternative path from the current node.*

b) (1pt) Demonstrate the key steps of this algorithm using a small example graph. *Note: The graph must be different from the Romania map used in the lecture.*

c) (0.5pt) Describe one limitation of the RBFS algorithm.
RBFS does not store all visited nodes, just know current path and best alternatives. As a result, it may revisit the expanded nodes multiple times if they reappear in different paths. This leads to increase computation times, especially in large or highly connected search spaces.