COSC1125/1127 Artificial Intelligence
Assignment 1 Search strategies
Due 11:59pm Monday, 25 March 2019

This is an individual assignment. It has **100 points** in total and is worth **10%** of the overall course grade. You may not collude with any other individual, or plagiarise their work. Students are expected to present the results of their own thinking and writing. Never copy another student's work (even if the other student "explains it to you first.") and never give your written work to others. Never copy from the web or any other resource. Remember you are meant to generate the solution to the questions by yourself. Suspected collusion or plagiarism will be dealt with according to RMIT University policy.

Questions 1 to 7 refer to the following 8-puzzle problems:

Problem #1	
Start	Goal
324	362
508	504
761	718

Problem #2	
Start	Goal
123	103
450	428
678	657

Problem #3	
Start	Goal
023	123
154	504
867	867

Problem #4	
Start	Goal
123	512
456	638
780	407

Problem #5	
Start	Goal
102	123
538	456
647	780

Problem #6	
Start	Goal
123	123
608	456
475	780

Electronic version of the tables used below can be obtained from the Canvas course page.

Assignment Questions

Q1. Depth-first vs breadth-first. Complete the following table of nodes in open and closed sets, and cost of the solution (the number of steps taken along the shortest path to the goal).

When a search generates more than 200 nodes in both the open set and the closed set, stop the applet. In such cases you can give the number of steps as "200+".

	Problem #1			Problem #2			Problem #3		
Search Type	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost
Depth-1st	.	.	.	.	.	.	.	.	.
Breadth-1st	.	.	.	.	.	.	.	.	.

Q2. Compare the performance of depth-first search and breadth-first search on each example. You need to account for any differences between the costs and number of nodes in terms of the properties of the different strategies (but you need only write a few lines).

Q3. Greedy vs. A* Complete the following table of nodes in open and closed sets, and cost of the solution.

	Problem #1			Problem #2			Problem #3		
Search Type	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost
Greedy + Tiles Out	.	.	.	.	.	.	.	.	.
Greedy + Euclidean	.	.	.	.	.	.	.	.	.
Greedy + City Block	.	.	.	.	.	.	.	.	.
A* Search + Tiles Out	.	.	.	.	.	.	.	.	.
A* Search + Euclidean	.	.	.	.	.	.	.	.	.
A* Search + City Block	.	.	.	.	.	.	.	.	.

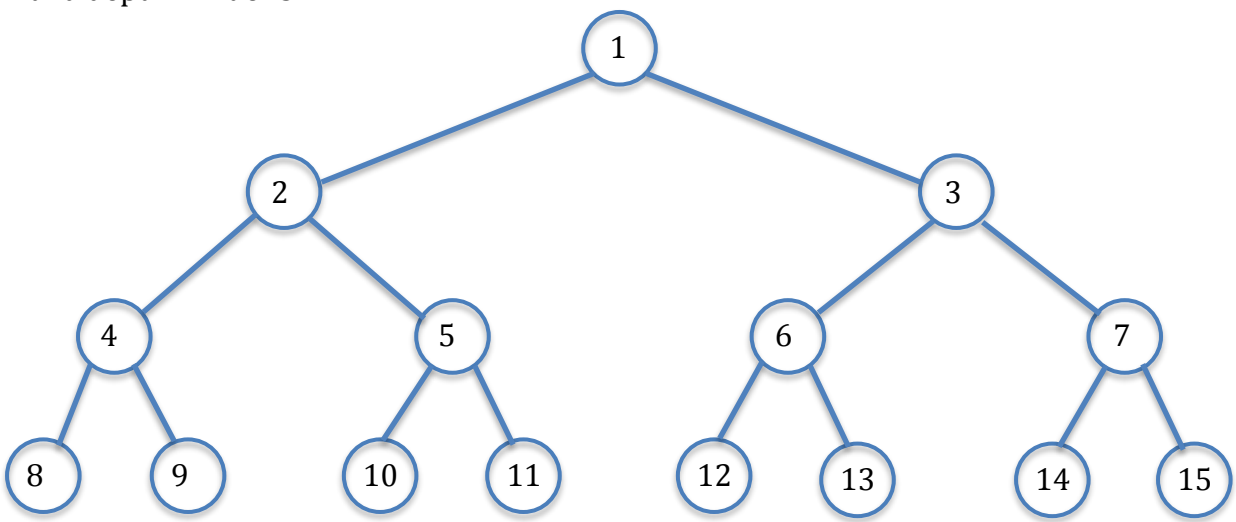
	Problem #4			Problem #5			Problem #6		
Search Type	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost	Open Set	Closed Set	Cost
Greedy + Tiles Out	.	.	.	.	.	.	.	.	.
Greedy + Euclidean	.	.	.	.	.	.	.	.	.
Greedy + City Block	.	.	.	.	.	.	.	.	.
A* Search + Tiles Out	.	.	.	.	.	.	.	.	.
A* Search + Euclidean	.	.	.	.	.	.	.	.	.
A* Search + City Block	.	.	.	.	.	.	.	.	.

Q4. What patterns do you notice when comparing Greedy search and A* search, and various combinations of heuristic measures?

Q5. What do the patterns from the previous question tell you about the different heuristics? Your answers should relate to the properties of A* search and the heuristics used, and particularly the properties of admissibility and informedness.

Q6. Design a heuristic for the 8-puzzle which is not admissible, i.e. it sometimes overestimates the cost. Show how it can lead to a suboptimal solution on a specific problem.

Q7. Consider a state space in which the initial state is 1 and the successor function for state n returns two states, numbers 2n and 2n+1. The figure below shows the state space diagram with a depth limit of 3.



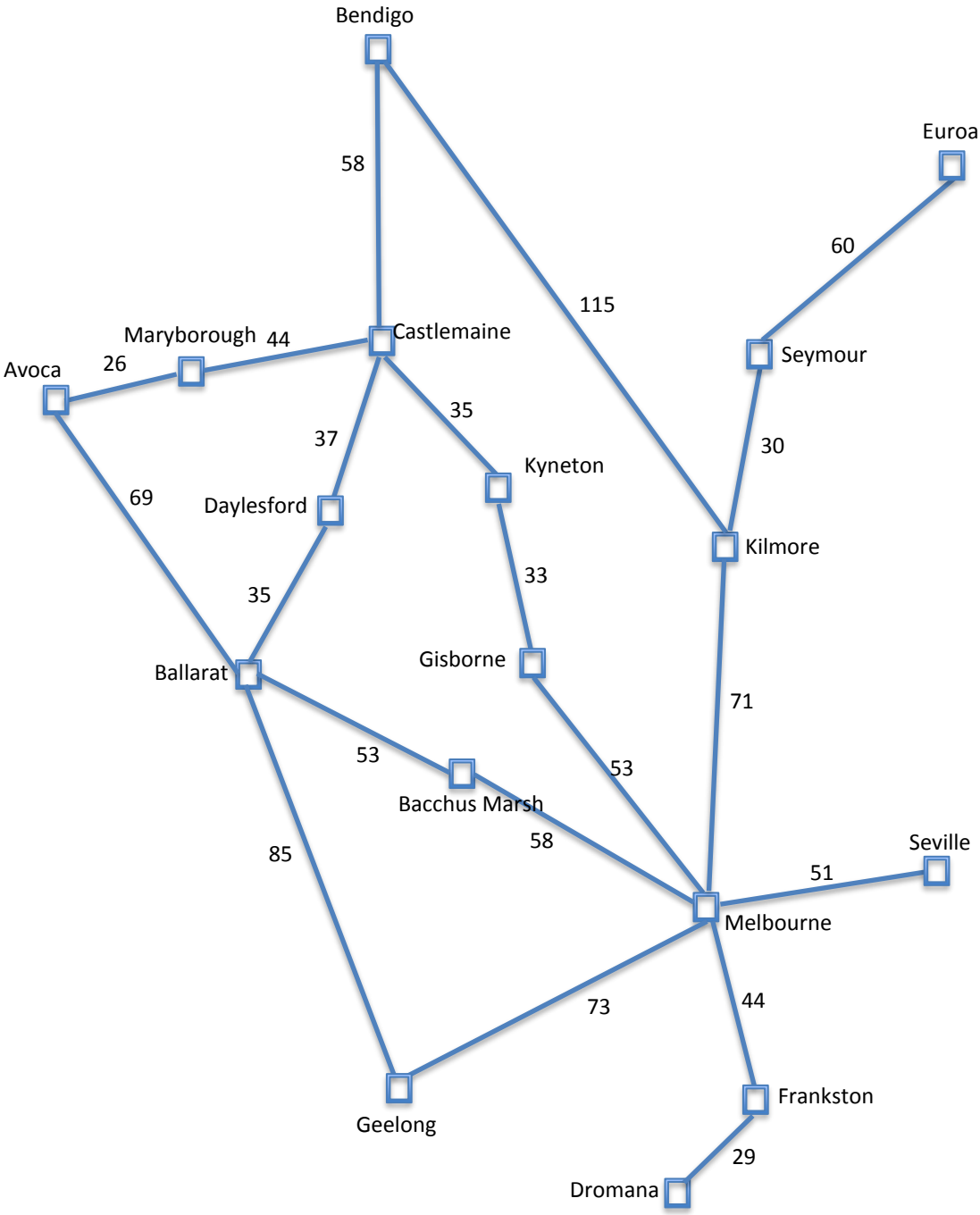
(a) Suppose the goal state is 12. List the order in which the nodes will be visited for breadth-first search, depth-limited search with depth limit 3 and iterative deepening.

(b) Would bi-directional search be appropriate for this problem? If so, describe how this would work. If not, explain why not.

(c) What is the branching factor in each direction of bi-directional search?

Q8. Trace the operation of A* search on the problem of getting from Bendigo to Melbourne using the straight-line distance heuristic (use the diagram and table below). Show the sequence of nodes that the algorithm will consider and the *f*, *g* and *h* scores for each node.

Straight-line distance to Melbourne	
Avoca	154
Bacchus Marsh	47
Ballarat	102
Bendigo	131
Castlemaine	106
Daylesford	88
Dromana	58
Euroa	131
Frankston	38
Geelong	66
Gisborne	50
Kilmore	58
Kyneton	81
Maryborough	137
Melbourne	0
Seville	45
Seymour	89



Submission Instructions
You are required to submit one **plain-text** file called *solutions.txt,* via the Canvas course page. Submissions that are not plain text will not be marked.

Late Submissions: 10% of the possible marks for this assignment will be deducted for each day late and assignments submitted 5 or more days late will not be marked.