

1. (14 points) This question is based upon Exercise 3.16 in our text. Read that exercise and answer the following questions. You do not have to do any programming for this question.
 - (a) Is the branching factor of this search tree the same at all levels? Find a formula that gives the number of nodes at depth $d \geq 1$. (I am looking for an exact answer, not the "rough" answer suggested in part d of the exercise. Hint: there is no expression at the root node, and there are two expressions at depth 1: 1 and n . Another hint: the text's answer is (much) too high.
 - (b) Which of the blind searches we covered would be suitable for this problem? Explain why. Explain why the other blind searches are not suitable.
 - (c) Comment on the suitability of blind search algorithms for solving this problem. Can you suggest other approaches?
2. (8 points) For this question, you will apply the A* algorithm to find a path in the graph below from the start node (S) to the goal node (G).

The heuristic we will use is the straight line distance to the goal node. Here is the distance from each node to the goal:

A	17.1	E	7.1	J	6.7
B	11.2	F	15.3	S	12.0
C	7.3	H	15.8		
D	4.1	I	8.6		

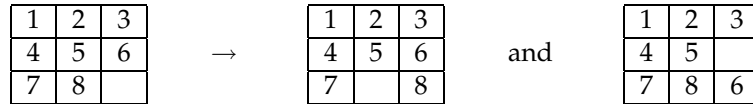
Show which nodes are on the queue at the following points in running the A* algorithm:

- (a) after the first node (S) is expanded (i.e. after its children have been added to the queue)
- (b) after the second node is expanded
- (c) after the third node is expanded
- (d) after the fourth node is expanded

Be sure to show $f(n)$, $g(n)$, and $h(n)$ for each node in the queue, and indicate which node has been expanded at each step.

3. (12 points) Suppose we represent the state of the 8 puzzle as a list of 9 elements: the first three correspond to the first row, the second three to the second row, etc. Write a function (`ep-children s`) that generates the successors to a state of the 8-puzzle problem.

For example:



```
(ep-children '(1 2 3 4 5 6 7 8 space))
==> ((1 2 3 4 5 space 7 8 6) (1 2 3 4 5 6 7 space 8))
```

You will find your `swap` procedure from Assignment 2 useful here.

4. (20 points) On the course home page, there will be a fairly efficient implementation of the A* algorithm. For this question, you will write two heuristics to solve the 8 puzzle. To test your heuristics with the A* implementation, you will need your `ep-children` function from Problem 3.
- (a) Write a function (`ep-manhattan state goal`) which returns the sum (over all tiles) of the manhattan distance from the current position of a tile to its position in the goal state.
 - (b) Write a function (`ep-heuristic state goal`) which implements your own heuristic for the 8 puzzle. I encourage you to design your own heuristic, but you can also implement a known heuristic or a variation on a known heuristic. Your score on this part will be proportional to the performance of your heuristic. (Performance will be a combination of the quality of solution and the computation time.)
 - (c) Explain your heuristic. Is it original, a variation on a known heuristic, or a known heuristic? Is this heuristic admissible? Is it monotonic?