1. Which of the following are true and which are false? Explain your answers.

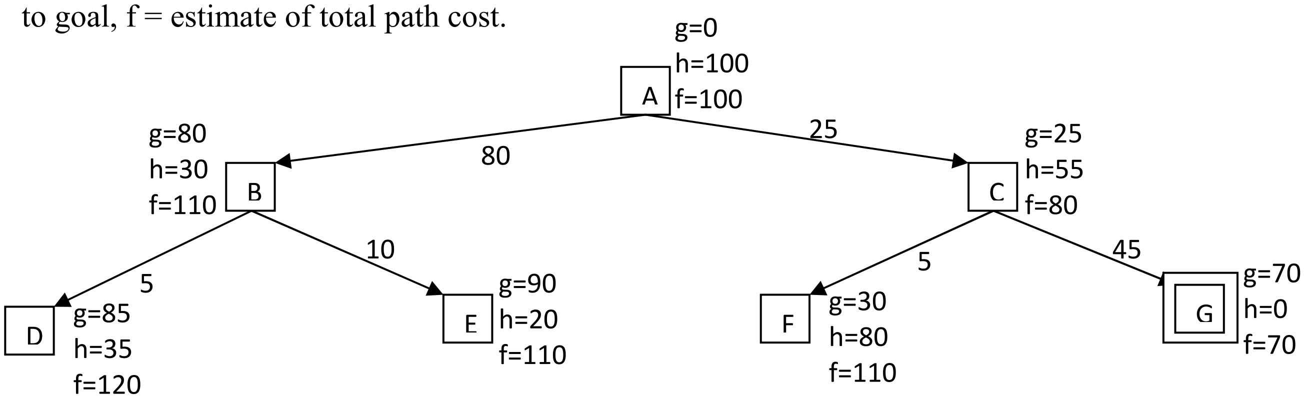
a. Depth-first search always expands at least as many nodes as A* search with an admissible heuristic.

b. h(n)=0 is an admissible heuristic for the 8-puzzle.

c. Breadth-first search is complete whenever the branching factor is finite, even if zero step costs are allowed.

d. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces; then Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

2. Use the following tree to indicate the order that nodes are expanded, for different types of search. Assume that A is the start node and G (double box) is the only goal node. Here, path costs are shown to the right of each path, g = cost of path so far, h = estimate of remaining cost to goal, f = estimate of total path cost.



a. Uniform-cost search.

b. Iterative deepening depth-first search.

c. Greedy best-first search.

d. A* search.

e. Is the heuristic h admissible?

f. Relabel the heuristic values h so that h is admissible but not consistent.

3. Recall that

* True path cost so far = g(n).

* Estimated cost to goal = h(n).

* Estimated total cost = f(n) = g(n) + h(n):

The following is a proof that A* tree search (queue sorted by f) is optimal if the heuristic h is admissible. The lines have been labelled A through G. Unfortunately, they have been scrambled.

Let ng be the first goal node popped off the queue. Let no be any other node on the queue. We wish to prove that no can never be extended to a path to any goal node that costs less than the path to ng that we just found.

A : true total cost of ng

B : ≤ f(no) // because queue is sorted by f

C : ≤ g(no) + true cost to goal from no // because h is admissible

D: =f(ng) // by definition of f with h(ng) = 0 since ng is a goal

E: =g(no) + h(no) // by definition of f

F: =g(ng) // because ng represents a complete path

G : = true total cost of no extended to a path to any goal node

Fill in the blanks with the letters B, C, D, E, F ,and G in the correct order to prove that the true total cost of ng ≤ true total cost of no. The first and last letters, A and G, have been done for you as an example.

A___ ___ ___ ___ ___G

4. Label the following as T (= True) or F (= False).

a. An admissible heuristic NEVER OVER-ESTIMATES the remaining cost (or distance) to the goal.

b. Best-first search when the queue is sorted by f(n) = g(n) + h (n) is both complete and optimal when the heuristic is admissible and the total cost estimate f(n) is monotonic increasing on any path to a goal node.

c. Most search effort is expended while examining the interior branch nodes of a search tree.

d. Uniform-cost search (sort queue by g(n)) is both complete and optimal when the path cost never decreases.

e. Greedy best-first search (sort queue by h(n)) is both complete and optimal when the heuristic is admissible and the path cost never decreases.

f. Beam search uses O(bd) space and O(bd) time.

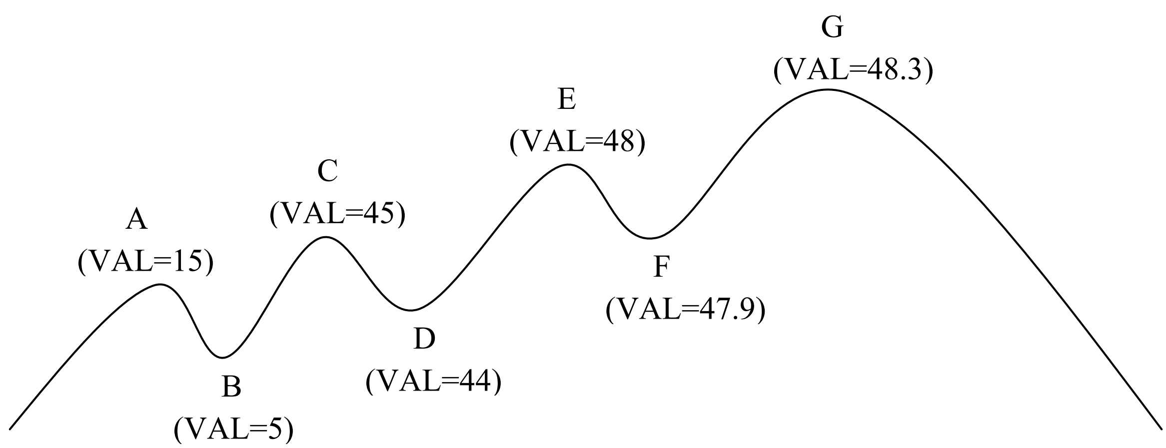
g. Simulated annealing uses O(constant) space and can escape from local optima.

h. Genetic algorithms use O(constant) space and can escape from local optima.

i. Gradient descent uses O(constant) space and can escape from local optima.

5. Perform Simulated Annealing search to maximize value in the following search space.

Recall that a good move (increases value) is always accepted (P = 1.0); a bad move (decreases value) is accepted with probability $P = e^{\Delta VAL/T}$, where $\Delta VAL = VAL(Next) - VAL(Current)$.



Use this temperature schedule:

Time Step	1–100	101–200	201–300
Temperature (T)	10	1.0	0.1

This table of values of *e* may be useful:

x	0.0	−1.0	−4.0	-4.3	−40.0	−43.0
e^x	1.0	≈0.37	≈0.018	≈0.014	$\approx 4.0*10^{-18}$	$\approx 2.1*10^{-19}$

a. (2 pts each, 56 pts total) Analyze the following possible moves in the search. The first one is done for you as an example.

Time	From	To	T	ΔVAL	ΔVAL/T	P
57	A	B	10	−10	-1	0.37
78	C	B				
132	C	B				
158	C	D				
194	E	D				
194	E	B				
238	E	D				
263	E	F				
289	G	F				
289	G	D				

b. At Time=100, is the search more likely to be in state A or in state C? (ignore E, G)

c. At Time=200, is the search more likely to be in state A, C, or E? (ignore G)

d. At Time=300, is the search more likely to be in state A, C, E, or G?