1. (20 points total, 5 pts each) Which of the following are true and which are false? Explain your answers. Unless stated otherwise, assume a finite branching factor, step costs $\geq \varepsilon > 0$, and at least one goal at a finite depth. You may be in either a tree or a graph.

a. (5 pts) Depth-first search always expands at least as many nodes as A search with an admissible heuristic.*

FALSE. Depth-first search may possibly, sometimes, BY GOOD LUCK, expand fewer nodes than A* search with an admissible heuristic. E.g., it is logically possible that sometimes, by good luck, depth-first search may march directly to the goal with no back-tracking.

b. (5 pts) h(n)=0 is an admissible heuristic for the 8-puzzle.

TRUE. h(n)=0 NEVER over-estimates the remaining optimal distance to a goal node.

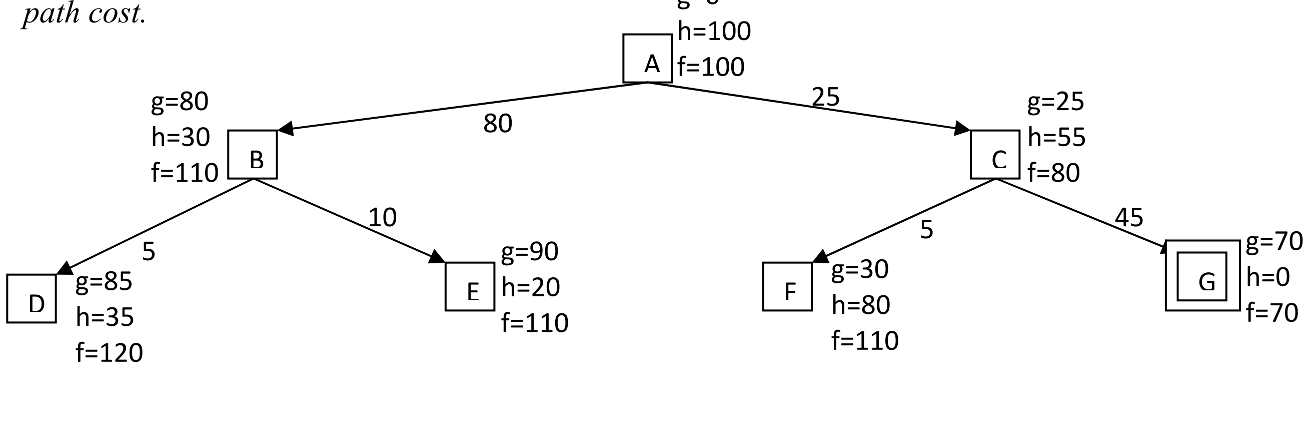
c. (5 pts) Breadth-first search is complete whenever the branching factor is finite, even if zero step costs are allowed.

TRUE, because if there exists a goal it occurs at finite depth *d* and will be found in $O(b^d)$ steps. "Complete" means "will find a goal when one exists" --- and does NOT imply "optimal," which means "will find a lowest-cost goal when one exists." Thus, the step costs are irrelevant to "complete."

d. (5 pts) 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.

FALSE. The Manhattan distance may over-estimate the optimal remaining number of moves to the goal because a rook may cover several squares in a single move. NOTE: If the path cost instead were the number of squares covered, then Manhattan distance would be admissible.

2. (20 points total, 5 pts off for each wrong answer, but not negative) 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. (5 pts) Uniform-cost search.

A, C, F, G

b. (5 pts) Iterative deepening depth-first search.

A; A, B, C; A, B, D, E, C, F, G.

c. (5 pts) Greedy best-first search.

A, B, E, D, C, G.

d. (5 pts) A search.*

A, C, G.

e. (5 pts) Is the heuristic h admissible?

NO. For example, h(A)=100, but the optimal cost to the goal node G is only 70. Thus, h(A) sometimes OVER-ESTIMATES the remaining optimal distance to G, and so is not admissible.

f. (5 pts) Relabel the heuristic values h so that h is admissible but not consistent.

There are many possibilities. One minimal change is to set h(A)=60 and h(C)=20. This is admissible because $h(A) = 60 \leq h^*(A) = 70$ and $h(C) = 20 \leq h^*(C) = 45$ (recall that h* is the true optimal remaining distance to the goal). It is not consistent because f(n) decreases along the path A-C-G, i.e., $f(A) = g(A)+h(A) = 0+60 = 60 > 45 = 25+20 = \text{cost}(A,C)+h(C) = g(C)+h(C) = f(C)$.

Any solution satisfying the following criteria should receive full credit:

Admissible:

$70 \geq h(A)$

$45 \geq h(C)$

Not consistent:

$h(A) > \text{cost}(A,C)+h(C) = 25+h(C)$

Admissible and not consistent:

$70 \geq h(A) > \text{cost}(A,C)+h(C) = 25+h(C)$

3. (20 points total, 5 pts off for each wrong answer, but not negative) 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*
F:	*=g(ng) // because ng represents a complete path*
D:	*=f(ng) // by definition of f with h(ng) = 0 since ng is a goal*
B :	*≤ f(no) // because queue is sorted by f*
E:	*=g(no) + h(no) // by definition of f*
C :	*≤ g(no) + true cost to goal from no // because h is admissible*
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 __F__ __D__ __B__ __E__ __C__ G

4. (20 points total, 5 pts off for each wrong answer, but not negative) Label the following as T (= True) or F (= False). Unless stated otherwise, assume a finite branching factor, step costs $\geq \varepsilon > 0$, and at least one goal at a finite depth. You may be in either a tree or a graph.

a. (5 pts) An admissible heuristic NEVER OVER-ESTIMATES the remaining cost (or distance) to the goal.

TRUE, by definition of admissible.

b. (5 pts) 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~~.

TRUE, because the search described is A* and the heuristic described is both admissible and consistent.

c. (5 pts) Most search effort is expended while examining the interior branch nodes of a search tree.

FALSE. Most search effort is expended while examining leaf node of the tree.

d. (5 pts) Uniform-cost search (sort queue by g(n)) is both complete and optimal when the path cost never decreases.

TRUE, because uniform-cost search is A* search with $h(n) = 0$, which is admissible.

e. (5 pts) 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.

FALSE. Your book gives a counter-example (Fig. 3.23, 3rd ed.; Fig. 4.2, 2nd ed.).

f. (5 pts) Beam search uses O(bd) space and O(bd) time.

FALSE. For a beam search in a tree using k nodes total, the space used is O(bk) and the time is O(bmk). For a beam search in a graph, the space is again O(bk) but it can waste time in loops.

g. (5 pts) Simulated annealing uses O(constant) space and can escape from local optima.

TRUE. The space is constant and it accepts bad moves with probability exp(-delta(Value)).

h. (5 pts) Genetic algorithms use O(constant) space and can escape from local optima.

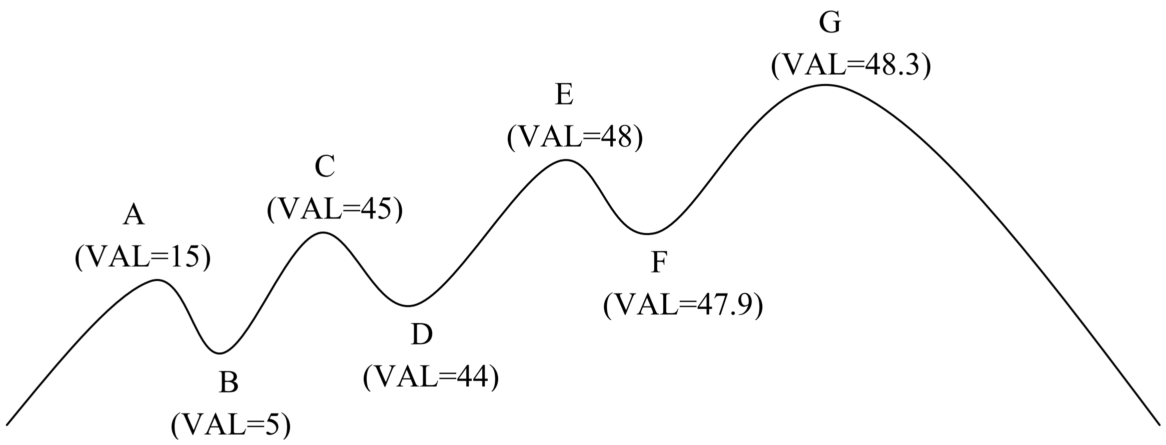
TRUE. The space is constant and it can accept bad moves by creating bad offspring.

i. (5 pts) Gradient descent uses O(constant) space and can escape from local optima.

FALSE. The space is constant, but it generally moves toward, and gets stuck on, a local optima.

5. (20 points total, 5 pts each) 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. (5 points total, 1 pt off for each wrong answer, but not negative)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*	10	−40	−4	≈0.018
132	*C*	*B*	1.0	−40	−40	$\approx 4.0*10^{-18}$
158	*C*	*D*	1.0	−1	−1	≈0.37
194	*E*	*D*	1.0	−4	−4	≈0.018
194	*E*	*B*	1.0	−43	−43	$\approx 2.1*10^{-19}$
238	*E*	*D*	0.1	−4	−40	$\approx 4.0*10^{-18}$
263	*E*	*F*	0.1	−0.1	−1	≈0.37
289	*G*	*F*	0.1	−0.4	−4	≈0.018
289	*G*	*D*	0.1	−4.3	−43	$\approx 2.1*10^{-19}$

b. (5 pts) At Time=100, is the search more likely to be in state A or in state C? (ignore E, G)

C.

c. (5 pts) At Time=200, is the search more likely to be in state A, C, or E? (ignore G)

E.

d. (5 pts) At Time=300, is the search more likely to be in state A, C, E, or G?

G.