Q1. Which of the following functions is suitable as a nonlinear activation function for training neural networks?

A. $g(x) = \min(x, 6)$

B. $g(x) = 2x + 1$

C. $\mathrm{g}(x) = 1 \text{ if } x > 0 \text{ and } -1 \text{ otherwise}$

D. $g(x) = x$

Q2. Which of the following is TRUE regarding the nonlinear activation functions (such as Sigmoid and ReLU) within neural networks?

A. They can speed up the gradient calculation in backpropagation, as compared to linear units.

B. They help to learn nonlinear decision boundaries.

C. They must always output positive values.

D. They are applied only to the hidden units.

Q3. If the neural network output is $y = (x^2 + w_1)^2 \cdot w_2$. What's the derivative/gradient for w_1?

A. 1

B. $x^2 + 1$

C. $w_2 \cdot (x^2 + 1)$

D. $w_2 \cdot 2(x^2 + w_1)$

Q4. What could be a possible scenario if we are running a breadth-first search (BFS) on an infinite tree-structured state-space graph? Suppose we do NOT have assumptions on the existence of the goal states or on the edge costs. Multiple answers are possible.

A. Gets stuck in an infinite loop.

B. Visits a finite number of nodes, then return one at random.

C. Finds the least-cost goal.

D. Finds a goal, but not necessarily the least cost goal.

Q5. What could be a possible scenario if we are running a uniform-cost search (UCS) on an infinite tree-structured state-space graph? Suppose we do NOT have assumptions on the existence of the goal state, but we assume that the edge cost is always $\geq \epsilon > 0$. Multiple answers are possible.

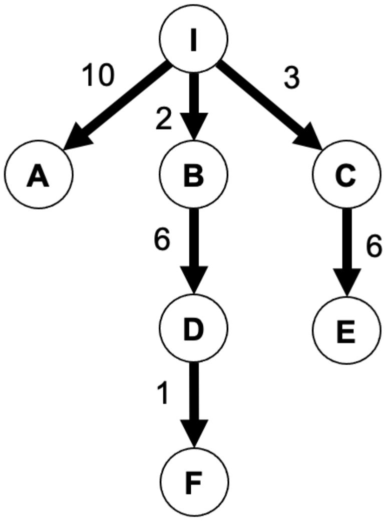
A. Gets stuck in an infinite loop.

B. Visits a finite number of nodes, then return one at random.

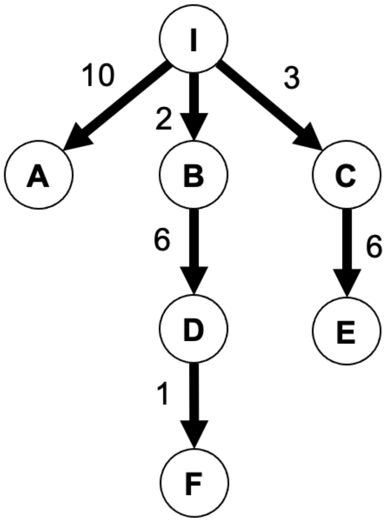
C. Finds the least-cost goal.

D. Finds a goal, but not necessarily the least cost goal.

Q6. Consider the following search tree. Initial state is I, and the goal state is D. Suppose we run BFS. Write down for each iteration, the node expanded, and the fringe at the end of the iteration. The fringe should be sorted, with the front on the left-hand side. For tie-breaking, following the dictionary order (e.g., A before B).



Q7. Consider the following search tree. Initial state is I, and the goal state is D. Suppose we run UCS. Write down for each iteration, the node expanded, and the fringe at the end of the iteration. The fringe need not be sorted. The expanded node, and the nodes in the fringe should be in the format of (node id, cost). For tie-breaking, following the dictionary order (e.g., A before B).


I
10
2
3
A
B
C
6
6
D
E
1
F