1. (a) (In the following I will use the notation \( \binom{n}{k_1, k_2, k_3} \) to mean \( \frac{n!}{k_1!k_2!k_3!} \). For the time being, allow states that occur after a goal state (ie a board position where there is both 3 X’s in a row and 3 O’s in a row is still counted). The number of possible states after exactly \( n \) plays can be calculated by a multinomial coefficient. For example, after 9 plays there are exactly \( \binom{9}{5, 4, 0} = \binom{9}{5} = \binom{9}{3} \) states, since there are 5 X’s to place in 9 locations, but the order doesn’t matter (and the remaining spots are filled with O’s). After 8 plays, there are \( \binom{9}{4, 4, 1} \) states, since there are 4 X’s to place, 4 O’s to place, and 1 square left empty. (Note that \( \binom{9}{4, 4, 1} = \binom{8}{4} \binom{8}{4} \), much like we are selecting 8 squares to fill in, then selecting 4 squares to place an X.) So, overall this gives:

\[
\binom{9}{5, 4, 0} + \binom{9}{4, 4, 1} + \binom{9}{4, 3, 2} + \binom{9}{3, 3, 3} + \binom{9}{3, 2, 4} + \binom{9}{2, 2, 5} + \binom{9}{2, 1, 6} + \binom{9}{1, 1, 7} + \binom{9}{1, 0, 8} + \binom{9}{0, 0, 9} = 6046
\]

However, now we need to subtract off the number of illegal “past goal” states, specifically all states that contain a two full rows (or columns) of X’s and O’s. First count all states where the XXX and OOO strings are columns on the Tic-Tac-Toe grid. For an empty board there are 3 empty columns and thus \( \binom{3}{1, 1, 1} = 6 \) possible ways to put the two columns down, and 3 remaining squares to fill. The 3 remaining squares can be all empty (1), contain a single X \( \binom{3}{2, 1, 0} = 3 \), contain one X and one O \( \binom{3}{1, 1, 1} = 6 \), or contain two X’s and one O \( \binom{3}{2} = 3 \). Thus there are \( 1 + 3 + 6 + 3 = 13 \) possible ways to fill the remaining 3 squares for a total of 13*6 states. Counting the states where the XXX and OOO strings are rows rather than columns gives another \( 13 \times 6 \) states for a total of \( 13 \times 12 = 156 \) states that are illegal. This gives a total of:

\[
6046 - 156 = 5890
\]

Note that it is a bit ambitious to get the count exactly. \( 3^9 \) is ok, \( 9! \) not as good (it counts multiple paths to the same state as multiple states, and doesn’t account for partially completed games!)

(b) The depth is 9. Yes the game tree contains all board positions in (a). This may not be true if you only counted an upper bound in part (a).

(c) Only drawing paths that aren’t symmetrically equivalent to another path:
(d)
(e) Branches marked with an $x$ above the edge are pruned: For left to right (ie bottom to top)
From right to left (ie top to bottom)

The optimal order would be to visit the branch in which X played in the center, first.

(f) To maximize pruning, putting the leaf nodes so that the subtrees under a MIN node are in increasing order (so MIN sees its best cases sooner), and the subtrees under a MAX node are in decreasing order (so MAX sees its best cases sooner). To minimize pruning, just do the opposite (ie decreasing subtrees under MIN and increasing subtrees under a MAX). Saying simply decreasing order (and increasing for minimizing pruning) is ok because they have this property, but aren’t as
general. Arbitrary trees cannot be made to have their leaves in monotonic order, 
but the subtree ordering property could always theoretically be satisfied in any 
tree.

2. (a) $D$, which gives an outcome of 8 at node $W$

(b) $O, Q, I, T, U, Y$

3. (a) $G : 3.5, F : 4, E : 4.5, D : 6, C : 3.5, B : 4.5, A : 4.5$

(b) $A (> -\infty), B (< \infty), D, H(4), D, I(8), D(6), B (< 6), E, J(2), K(7), E(4.5), B(4.5), A (> 
    4.5), C (< \infty), F, L(3), M(5), F(4), C (< 4), G, N(1), O(6), G(3.5), C(3.5), A(4.5)$

(c) $G$ and its children could have been pruned, since after evaluating $F$ we know 
    $C < 4$, and we already have $B = 4.5$.

(d) $K$ with probability .5 and $J$ with probability .5.

(e) No, he was maximizing his expected gain, over many games MAX will do better, 
    he was just unlucky.