1 Uninformed Search

Consider the search tree in Fig. 1. Assume that the exploration of the children of a particular node proceeds from the left to the right for all search methods in this section.

**Question 1 (4 pts):** Give an example of when breadth-first search (BFS) and depth-first search (DFS) have the same time complexity. Pick on node number from the tree above as an example.

**Question 2 (4 pts):** Give an example when depth first search is suboptimal. Pick two node numbers as goal nodes as an example.

**Question 3 (4 pts):** What limitation in BFS does iterative deepening search overcome?

**Question 4 (4 pts):** Why is the space complexity of BFS $O(b^{d+1})$, not $O(b^d)$, where $b$ is the branching factor and $d$ is the goal depth?

**Question 5 (4 pts):** Can depth limited search become incomplete in the case of the finite search tree above? If so, give an example. If not, explain why not.

2 Informed Search

**Question 6 (10 pts):** Manually conduct greedy best-first search on the graph below (Fig. 2), with initial node a and goal node m. Actual cost from node to node are shown as edge labels. The heuristic function value for each node is shown in a separate table to the right. Show:

1. Node list content at each step
2. Node visit order
3. Solution path
4. Cost of the final solution.

Note: Assume that you are not allowed to go back immediately to where you came from (i.e., no backtracking).

Question 7 (10 pts): (1) Repeat the problem right above with A* search. (2) In addition, show the $f(n)$ value for all nodes expanded. (3) Which one gives a shorter solution: Greedy best-first or A*? Note: Note that the same node can appear in the node list with a different $f(n)$ value, depending on the path taken.

Question 8 (12 pts): Explain why A* is optimal, in the general case. Explain in terms of an arbitrary node $n$ on the path to an optimal goal $G_1$, and a separate suboptimal goal $G_2$. 

<table>
<thead>
<tr>
<th>Node</th>
<th>h(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>22</td>
</tr>
<tr>
<td>b</td>
<td>22</td>
</tr>
<tr>
<td>c</td>
<td>41</td>
</tr>
<tr>
<td>d</td>
<td>14</td>
</tr>
<tr>
<td>e</td>
<td>10</td>
</tr>
<tr>
<td>f</td>
<td>10</td>
</tr>
<tr>
<td>g</td>
<td>20</td>
</tr>
<tr>
<td>h</td>
<td>40</td>
</tr>
<tr>
<td>i</td>
<td>22</td>
</tr>
<tr>
<td>j</td>
<td>31</td>
</tr>
<tr>
<td>k</td>
<td>14</td>
</tr>
<tr>
<td>l</td>
<td>19</td>
</tr>
<tr>
<td>m</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2: Informed Search.
3 Game Playing

3.1 Minmax Search

Question 9 (4 pts): Using the following figure 3, use minmax search to assign utility values for each internal node (i.e., non-leaf node) and indicate which path is the optimal solution for the MAX node at the root of the tree. Assume you explore the successors from left to right.

![Game Tree](image)

Figure 3: Game Tree. Solve using minmax search.

3.2 $\alpha - \beta$ pruning

Question 10 (8 pts): Using the following figure 4, use $\alpha - \beta$ pruning to (1) assign utility values for each internal node (i.e., non-leaf node) and indicate which path is the optimal solution for the MAX node at the root of the tree. (2) For each node, indicate the final $\alpha$ and $\beta$ values. (Note that initial values at the root are $\alpha = -\infty, \beta = \infty$.) (3) For each cut that happens, draw a line to cross out that subtree.

![Game Tree](image)

Figure 4: Game Tree. Solve using $\alpha - \beta$ pruning. This tree is the same as figure 3.

Question 11 (6 pts): In Minmax search, we used a depth-first exploration through the use of recursion. We know that Minmax gives an optimal solution, however, we also know that depth-first search is suboptimal. Explain why Minmax gives an optimal solution even when it is using a depth-first exploration.
4 Logic and Theorem Proving

4.1 Normal forms

In all of the problems in this section, show each step of the derivation and indicate which axioms (or other rules) you used: For example, *distributive law; by definition, etc.*

**Question 12 (2 pts):** Convert \( \neg(P \rightarrow \neg S) \lor (\neg(S \rightarrow (Q \rightarrow R))) \) into conjunctive normal form.

**Question 13 (2 pts):** Convert \( \neg T \rightarrow (R \land (\neg(P \land Q) \rightarrow \neg S)) \) into disjunctive normal form.

**Question 14 (8 pts):** Convert the following into prenex normal form, and then into conjunctive normal form, and then skolemize.

1. \( \neg \forall x (\neg P(x) \lor \neg (\exists y, Q(x, y))) \)
2. \( \neg \exists x ((\exists y Q(x, y)) \rightarrow P(x)) \)

4.2 Theorem proving

**Question 15 (18 pts):** Show that \( \exists x (D(x) \land C(x)) \) is a logical consequence of the following, using resolution.

1. \( \neg E(x) \lor V(x) \lor S(x, f(x)) \)
2. \( \neg E(x) \lor V(x) \lor C(f(x)) \)
3. \( E(a) \)
4. \( D(a) \)
5. \( \neg S(a, y) \lor D(y) \)
6. \( \neg D(x) \lor \neg V(x) \)

**Hint:** first, transform the problem into a set of clauses, and then follow the resolution steps. Don’t forget the negate the conclusion.