1. (25 pts total, -5 pts for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is Max's turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

1.a. Fill in each blank square with the proper mini-max search value.

1.b. What is the best move for Max? (write A, B, or C) ____ B ____

See Section 5.2.1.

2. (25 pts total, -5 for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning.

See Section 5.3.

Red lines indicate where in the tree pruning occurred. You are not obliged to provide the red lines — only to cross out pruned leaf nodes.

*** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ***
3. (50 pts total, 10 pts each)

Execute Tree Search through this graph (i.e., do not remember visited nodes, so repeated nodes are possible). Step costs are given next to each arc. Heuristic values are given next to each node (as \( h=x \)). The successors of each node are indicated by the arrows out of that node. Successors are returned in left-to-right order. (Note: \( C \) is a successor of itself).

For each search strategy below, indicate the order in which nodes are expanded (i.e., to expand a node means that its children are generated), ending with the goal node that is found. The first one is done for you as an example.

3.a. DEPTH FIRST SEARCH.

\[ S \rightarrow A \rightarrow D \rightarrow G_1 \]

See Section 3.4.3 and Fig. 3.17.

DFS does the Goal-test before the child is pushed onto the queue. The goal is found when \( D \) is expanded.

3.b. (10 pts, -3 for each wrong answer, but not negative) UNIFORM COST SEARCH.

\[ S \rightarrow C \rightarrow B \rightarrow A \rightarrow F \rightarrow C \rightarrow E \rightarrow G_1 \]

See Section 3.4.2 and Fig. 3.14.

UCS does goal test when node is popped off queue.

3.c. (10 pts, -3 for each wrong answer, but not negative) GREEDY (BEST-FIRST) SEARCH.

\[ S \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow \text{etc.} \]

See Section 3.5.1 and Fig. 3.23.

C always has lower \( h(=11) \) than any other node on queue.

3.d. (10 pts, -3 for each wrong answer, but not negative) ITERATED DEEPENING SEARCH.

\[ S \rightarrow S \rightarrow A \rightarrow B \rightarrow C \rightarrow S \rightarrow A \rightarrow D \rightarrow G_1 \]

See Sections 3.4.4-5 and Figs. 3.18-19.

IDS does the Goal-test before the child is pushed onto the queue. The goal is found when \( D \) is expanded.

3.e. (10 pts, -3 for each wrong answer, but not negative) A* SEARCH.

\[ S \rightarrow C \rightarrow B \rightarrow A \rightarrow F \rightarrow C \rightarrow E \rightarrow G_2 \]

See Section 3.5.2 and Figs. 3.24-25.

A* does goal test when node is popped off queue.

3.f. (10 pts, -3 for each wrong answer, but not negative) OPTIMALITY.

Did Uniform Cost Search find the optimal goal? Yes  Why or why not? Step costs are \( \geq \varepsilon > 0 \)

Did A* Search find the optimal goal? No  Why or why not? heuristic is not admissible (at \( D \))

OK to say “Heuristic overestimated” or “Heuristic is too high.”
3.e. A* SEARCH. Represent nodes as state/h/g/f.

Initial Queue: (S/10/0/10)

Popped Node: (S/10/0/10)

Children (left-to-right): (A/13/5/18) (B/12/4/16) (C/11/2/13)

Queue (sorted): (C/11/2/13) (B/12/4/16) (A/13/5/18)

Popped Node: (C/11/2/13)

Children: (C/11/13/24) (F/13/10/23)

Queue: (B/12/4/16) (A/13/5/18) (F/13/10/23) (C/11/13/24)

Popped Node: (B/12/4/16)

Children: (E/11/14/25)

Queue: (A/13/5/18) (F/13/10/23) (C/11/13/24) (E/11/14/25)

Popped Node: (A/13/5/18)

Children: (D/16/15/31)

Queue: (F/13/10/23) (C/11/13/24) (E/11/14/25) (D/16/15/31)

Popped Node: (F/13/10/23)

Children: (G3/0/34/34)

Queue: (C/11/13/24) (E/11/14/25) (D/16/15/31) (G3/0/34/34)

Popped Node: (C/11/13/24)

Children: (C/11/24/35) (F/13/21/34)

Queue: (E/11/14/25) (D/16/15/31) (G3/0/34/34) (F/13/21/34) (C/11/24/35)

Popped Node: (E/11/14/25)

Children: (G2/0/29/29)

Queue: (G2/0/29/29) (D/16/15/31) (G3/0/34/34) (F/13/21/34) (C/11/24/35)

Popped Node: (G2/0/29/29)

Children: Goal test succeeds.