Search Algorithms Assignment

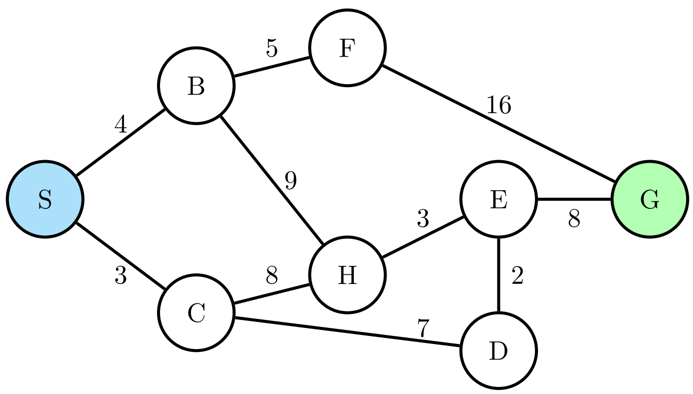
K.N.Toosi University of Technology
Artificial Intelligence

Spring 2025

Part I
Practical Assignment

Consider the following state-space graph, where S is the start state and G is the goal state. The cost of each edge is given on the graph, and the edges are bidirectional. Using the heuristic table provided, answer the following questions.

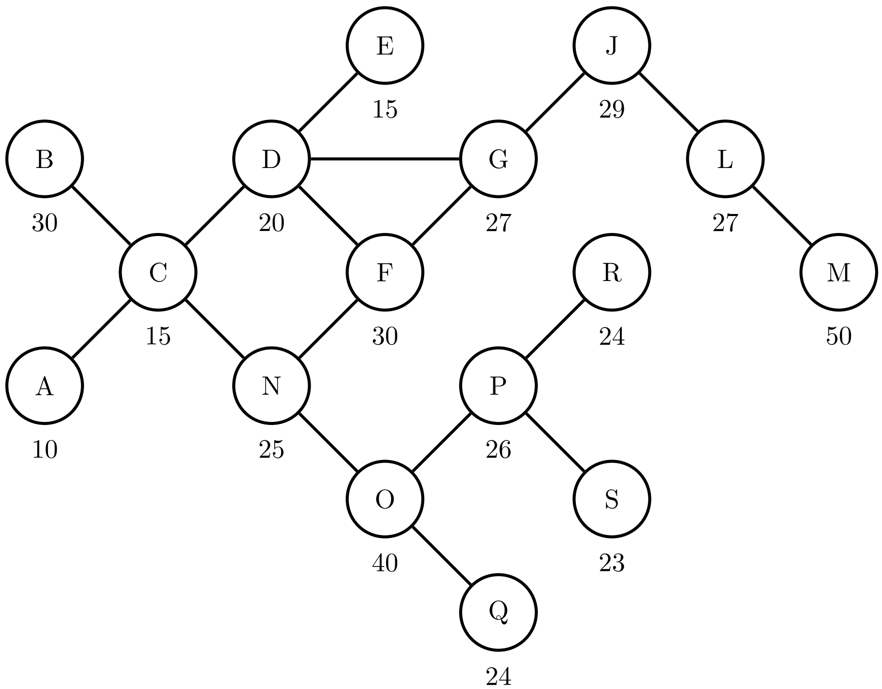
Node	**h**
S	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



- Run the following algorithms on the given graph, visualize the search tree and construct the *frontier* and *explored* queues:
  - Breadth-First Search (BFS)
  - Uniform Cost Search (UCS)
  - Iterative Deepening Search (IDS)
  - A* when $h(E) = 6$
- For what values of $h(E)$ is the given heuristic both admissible and consistent?

Part II
Practical Assignment

Answer the following questions according to this graph:



- Perform the local beam search algorithm with $k = 2$ on the figure below, starting from nodes E and A to reach the node with the highest heuristic. The heuristic of each node is written next to it. Show at each step which two nodes we are at. Does the algorithm eventually reach the goal node M?
- Does the local beam search algorithm, assuming $k = 3$ and starting from nodes P, E and A, reach the goal node M?
- For $k = 1$ and $k = \infty$, what algorithms does this algorithm transform into?

Part III

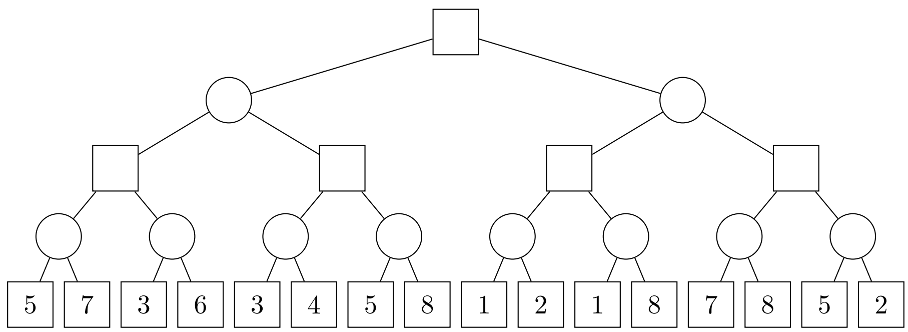
Practical Assignment

Answer the following questions regarding the **Simulated Annealing** algorithm.

- If, during the execution of the algorithm, we want to increase the speed of convergence , what changes would you suggest to the cooling schedule? Why?
- If the Simulated Annealing algorithm frequently gets stuck in local extrema, what solution would you propose to improve the algorithm's performance? Why?
- If we know that the search space has no local extrema in certain regions, which algorithm would you recommend: Simulated Annealing or Hill Climbing? Why?
- If we run the algorithm with a very high initial temperature and do not reduce the temperature until the end, which algorithm does Simulated Annealing resemble?

Part IV
Practical Assignment

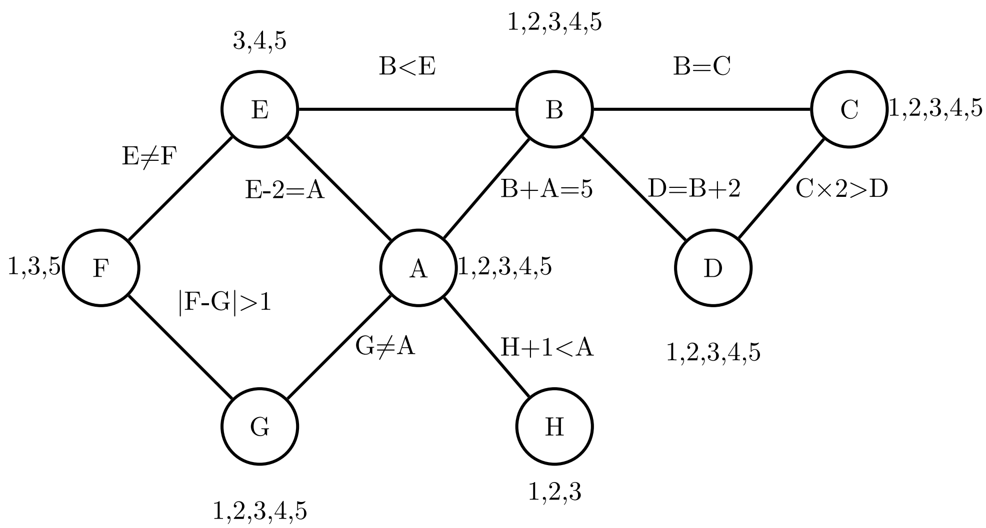
Consider the following tree. The max nodes are represented by squares, and the min nodes are represented by circles. The range of scores is $[1, 8]$.



- Execute the minimax algorithm on the given tree.
- Execute the alpha-beta pruning algorithm on the given tree and determine which nodes are pruned and for what reasons.

Part V
Practical Assignment

Given the CSP problem with the constraint graph below, transform this constraint graph into a tree structure by removing the fewest possible number of nodes, and obtain the solution to the problem based on it (when assigning values to variables, smaller values are given higher priority).



Part VI

Implementation Assignment: Chess Game Development with Minimax AI

1 Objective

In this assignment, you will develop a computer vs human chess game using `pygame` for visualization and `python-chess` for move validation. You will implement an AI opponent using the Minimax algorithm with a limited depth and an evaluation function to assess board positions. Finally, you will write a report analyzing the impact of search depth and evaluation functions on AI performance.

2 Requirements

2.1 Game Implementation

- Use `pygame` to render a chessboard and display the pieces.
- Allow a human player (White) to make legal moves by clicking squares.
- Implement an AI opponent (Black) using the Minimax algorithm with a limited depth.
- Use `python-chess` to determine valid moves and apply them.
- Detect checkmate, stalemate, or draw conditions and display a message.

2.2 Minimax AI

- Implement Minimax with a limited search depth.
- Create an evaluation function (e.g., material count).

2.3 Report

- Explain how `python-chess` was used for move validation.
- Discuss the Minimax algorithm, the evaluation function, and move ordering.
- Analyze how increasing/decreasing depth affects AI strength and performance.
- Analyze the impact of pruning on the efficiency of AI performance and the selection of moves.
- Include experiments where the AI plays against itself at different depths.

3 Submission

Submit the following files:

- Python code implementing the game.
- Report analyzing AI behavior with different depths and evaluation functions.

4 Resources

- `python-chess` documentation
- `Pygame` documentation
- Minimax algorithm

Note

Any attempt to use AI tools for generating the code is strictly prohibited. Students will be asked to present and explain their code during a class session.