

***Exercise 1:***

|  |  |
| --- | --- |
| State | h(n) |
| Z | 370 |
| A | 360 |
| T | 320 |
| C | 160 |
| R | 190 |
| P | 80 |
| G | 0 |
| F | 170 |
| S | 250 |
| O | 385 |

Apply A\* algorithm:

Step 1:

- Start with node A  
- Node Z, T, S can be reach from node A

A\* Algorithm calculates f(Z), f(T) and f(S).

* A->Z f(Z) = g(Z) + h(Z) = 75 + 370 = 445 (hold)
* A->T f(T) = g(T) + h(T) = 110 + 320 = 430 (hold)
* A->S f(S) = g(S) + h(S) = 140 + 250 = 390 (accept)

Since f(S) < f(T) < f(Z), so it decides to go to node S.

Path: A->S

Step 2:

- Node F and R can be reach from node S

A\* Algorithm calculates f(F), f(R).

* A->S->F f(F) = g(F) + h(F) = 230 + 170 = 400 (accept)
* A->S->R f(R) = g(R) + h(R) = 220 + 190 = 410 (hold)

Since f(F)<f(R), so we decide to go to node F

Path: A->S->F

Step 3:

- Node G can be reach from node F

A\* Algorithm calculates f(G).

* A->S->F-G f(G) = g(G) + h(G) = 350 + 0 = 350 ( still more optimized compare to all the hold path above so we accept)

We decide to go to node G

Path: A->S->F->G

This is the most optimal path from node A to G

***Exercise 2:***

Model-Based Chess-Playing Agent Architecture:

1. Environment Model (World Model):
   * A chess board with chess pieces

     Description automatically generatedBoard Representation: The agent maintains an internal representation of the chessboard, usually a 2D array, where each square holds information about the pieces on it.

Figure : Example of a 2D chessboard with 2D array

* + Chess Rules: The agent already has knowledge of the rules of chess, including special moves.

1. Sensors:
   * Perception: Sensors sends current send the current state of the chess board to the agent, specifically the location and type of each pieces.
2. Internal State:
   * Position Evaluation: The agent has an evaluation function that assigns values to different aspects of the board, such as piece material, piece mobility, king safety, and positional advantage.
   * Search Algorithm: The agent will use a search algorithm (example: Min-max Searching) to stimulate possible futures moves and explore different game states.
   * Chess Strategies and Tactics: The agent's internal state includes a knowledge base of chess strategies that can apply to current situation.
3. Action Selection:
   * Decision-Making: Based on the current position evaluation and the results of the search algorithm, the agent selects the best move to make by considering the consequences of various possible moves and the impact of those moves for future states.
4. Actuators:
   * Action Execution: The agent send out the moves to the chessboard, specifically the current location and the next location of the desired piece. (Example: Send the piece in location 2-a to 4-a)

A diagram of a business process

Description automatically generated

Detail description:

1. Enviroment model: the environment will be a chessboard with all the pieces on the board.
2. Sensors: This component will collect current state of the game, including the types and locations of the pieces.
3. Internal state: This state is in charge of analyzing position, search for possible moves and find stratregies or tactics and then send out these information.
4. Action selection: This components receive information from the internal state then select action and send it to the actuator
5. Actuator: This part will execute action to the environment