**CPSC 481- Artificial Intelligence**

**Handout – Intro, StateSpaceSearch & Heuristics**

# Short answer questions

1. What is an intelligent agent?

Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators

Ex: Human – Sensors: Eyes, ears, other organs; Actuators: hands, legs, mouth, other body parts

Ex: Robot – Sensors: Cameras, infrared range finders; Actuators: Various motors

1. Create and justify your own definition of artificial intelligence.

Intelligence utilized by a system that makes decisions based on its environment

1. What is the general problem-solving process?
2. A problem to be solved
3. Represent the problem in state space (conceptual)
4. Represent the state space represented problem in graph (symbolic)
5. Represent the graph represented problem using data structures and search algorithm (implementation using computer)
6. solution

1. What is data-driven search? For what types of problems do we want to use data-driven search?

Aka forward search. Data-driven search is using the knowledge and constraints found in each state of the problem to guide search by applying rules/methods to produce new states until it finds a goal state/solution. It is better - when all or most of the data are given in the initial problem state

- when there are large number of potential goals, but there are only a few way to use the facts and given information of a particular problem instance

- when it is difficult to formulate a goal or hypothesis

1. If a machine passes the Turing test, can the machine solve most AI problems? Why or why not?

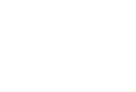
No. just because an agent can imitate intelligent behavior, it does not mean that the agent is inherently intelligent

1. If the opponent makes a mistake, will the Mini-max still work?

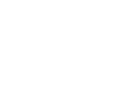
yes

1. What are the different types of environment for agents?

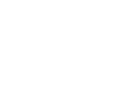
# Search



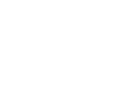
S



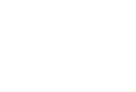
A



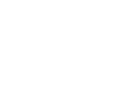
B



C



D



G



1



3



1



1



2



3



12



3

Answer the following questions about the search problem shown above. S is the start state, G is the goal state. Break any ties alphabetically. For the questions that ask for a path, please give your answers in the form `S - A - D - G.'

1. What path would breadth-first graph search return for this search problem?

1. What path would depth-first graph search return for this search problem?

1. What path would A\* graph search, using h1 heuristic as shown in the table below, return for this search problem?

|  |  |  |
| --- | --- | --- |
| State | h1 | h2 |
| S | 5 | 4 |
| A | 3 | 2 |
| B | 6 | 6 |
| C | 2 | 1 |
| D | 3 | 3 |
| G | 0 | 0 |

1. Consider the heuristics for this problem shown in the table above.

|  |  |  |  |
| --- | --- | --- | --- |
| i. Is h1 admissible? |  | Yes | No |
| ii. Is h2 admissible? |  | Yes | No |

iii.

Is h1 consistent (monotone/local admissible), why or why not?

iv. h1 and h2, which one is more informed and why?

# Heuristics

The sliding-tile puzzle consists of three black tiles, three white tiles, and an empty space in the configuration shown in the following figure. The puzzle has two legal moves with associated costs: A tile may move into an adjacent empty location. This has a cost of 1. A tile can hop over one or two other tiles into the empty position. This has a cost equal to the number of tiles jumped over. The goal is to have all the white tiles to the left of all the black tiles. The position of the blank is not important.

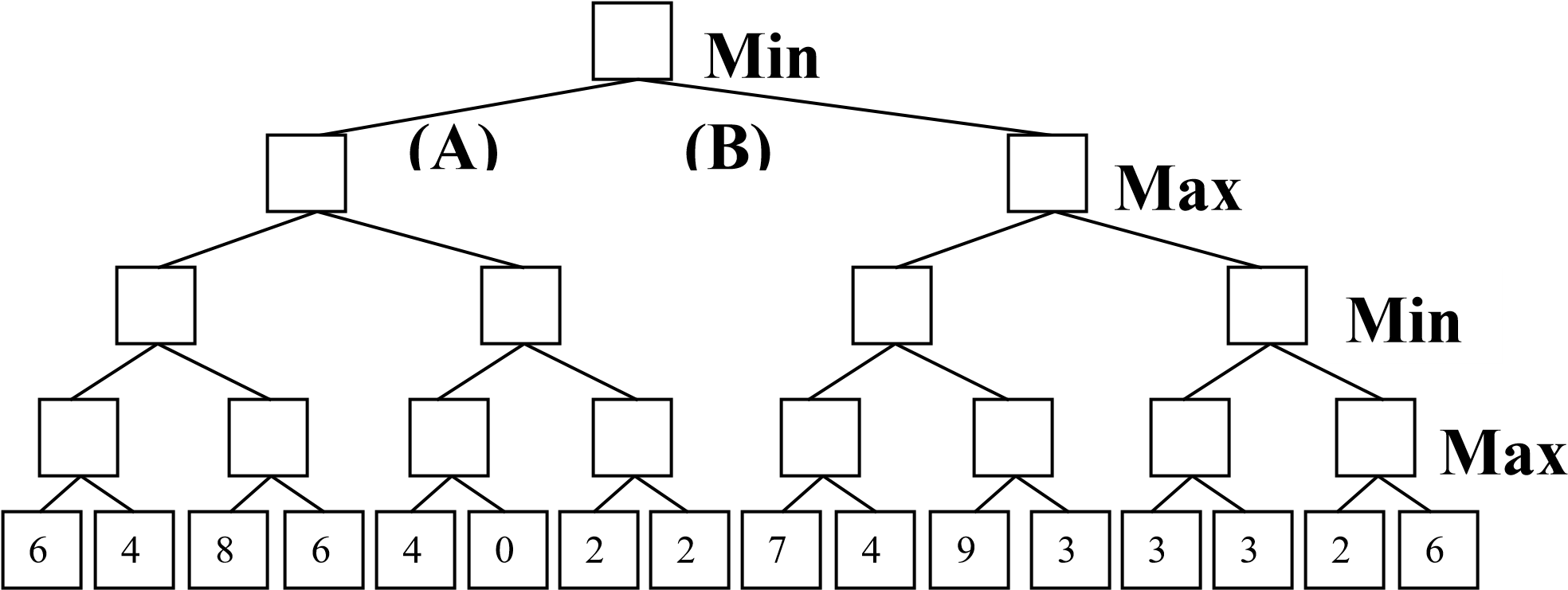
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **B** | **B** | **B** |  | **W** | **W** | **W** |

1. (6 pts) Draw the first three levels (level 0 - 2) of the state space graph.

1. (8 pts) Propose an admissible heuristic for solving this problem. You are to describe and define h(n) and explain why it is admissible.

# Game Tree

1. The game tree below illustrates a position reached in the game. It is MIN's turn to move. Inside each leaf node is the estimated score of that resulting position returned by the heuristic static evaluator. FILL IN EACH BLANK SQUARE WITH THE PROPER VALUE ACCORDING TO MINI-MAX SEARCH.



1. (2 pts) What is MIN's best move (write A or B)

1. This is the same tree and conditions as above. CROSS OUT EACH LEAF NODE THAT WILL NOT BE EXAMINED BECAUSE IT IS PRUNED BY ALPHA-BETA PRUNING. You do NOT need to cross out branches. You do not need to indicate the branch node values again.

