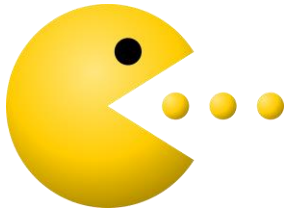
FACULTY OF
INFORMATION TECHNOLOGY
AND ELECTRICAL ENGINEERING

UNIVERSITY
OF OULU

Artificial Intelligence (A.I.)
Project #2

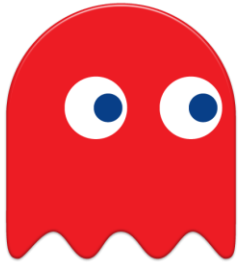
Miika Malin

Original slides created by Mohammad Tavakolian in 2021



In Project 1, we implemented search algorithms in a efficient and optimal way.

Using generalized search methods, the agents could determine the best possible plan and them simply execute it to arrive at a goal.



Now, let's consider scenarios where our agents have one or more adversaries who attempt to keep them from reaching their goals.

Our agents can no longer run the search algorithms we have already learned to formulate a plan as we typically don't deterministically know how our **adversaries** will plan against us and respond to our actions.

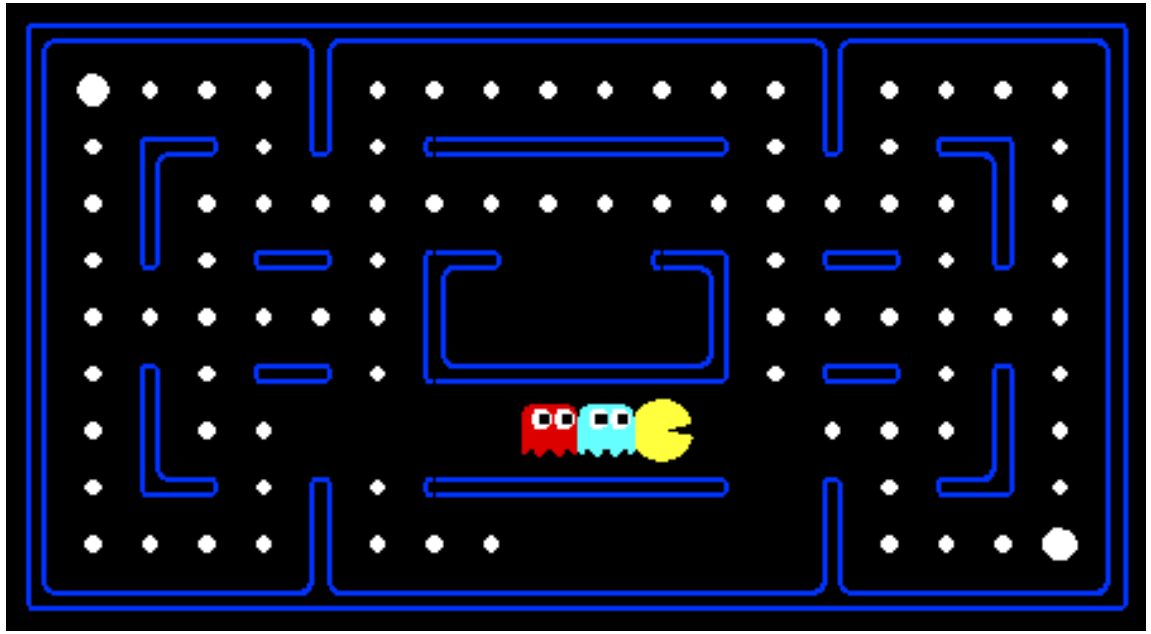
We need to run a new class of algorithms that yield solutions to adversarial search problems (games).

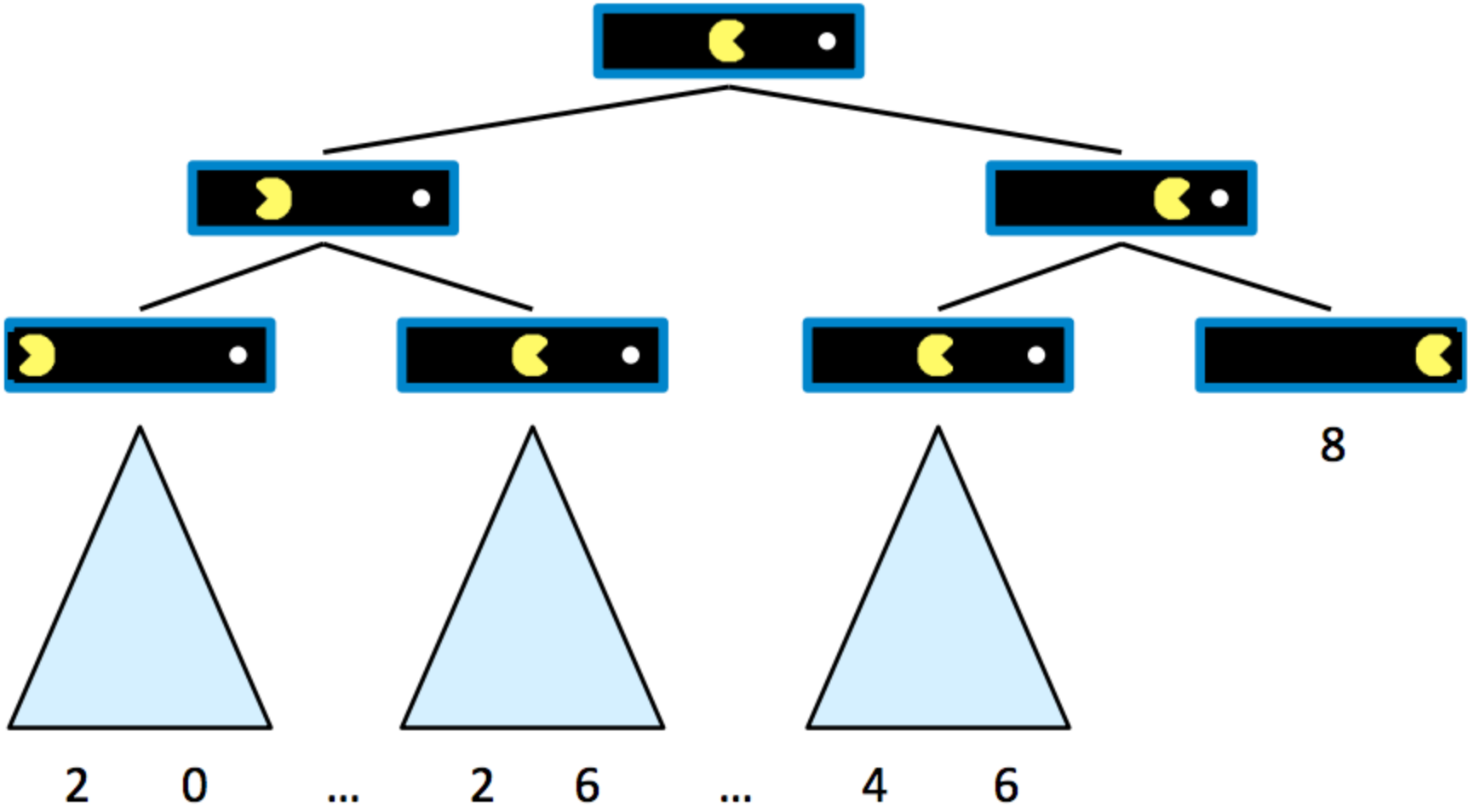
Adversarial Search

- Pacman, now with ghosts
- Minimax, Expectimax, Evaluation

You will design agents for the classic version of Pacman, including ghost.

Help Pacman agent to eat dots, while avoiding the ghosts.

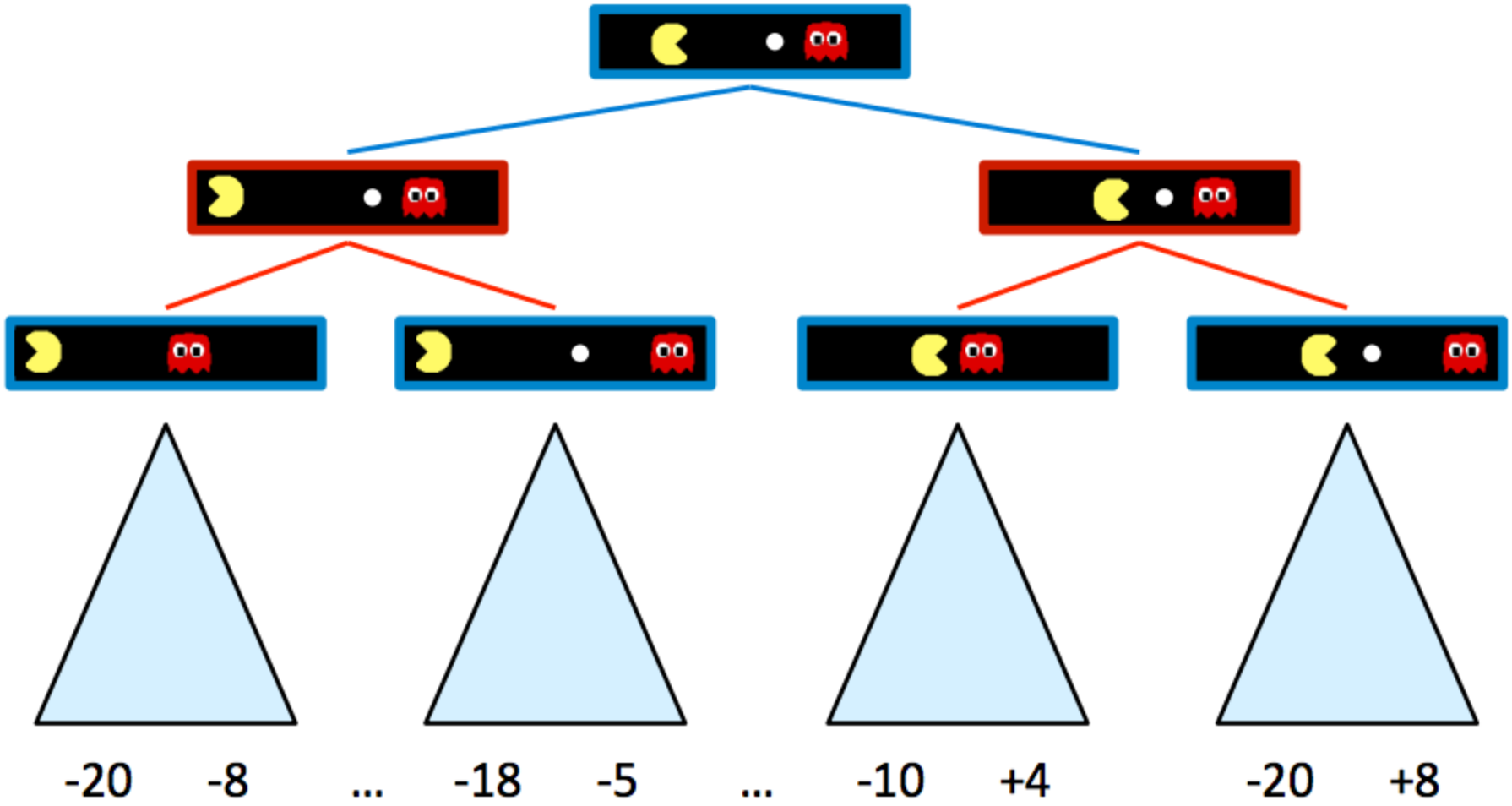




ent behaves optimally, and

od.

Note: Children of a state are successor states just as in search trees for normal search problems.



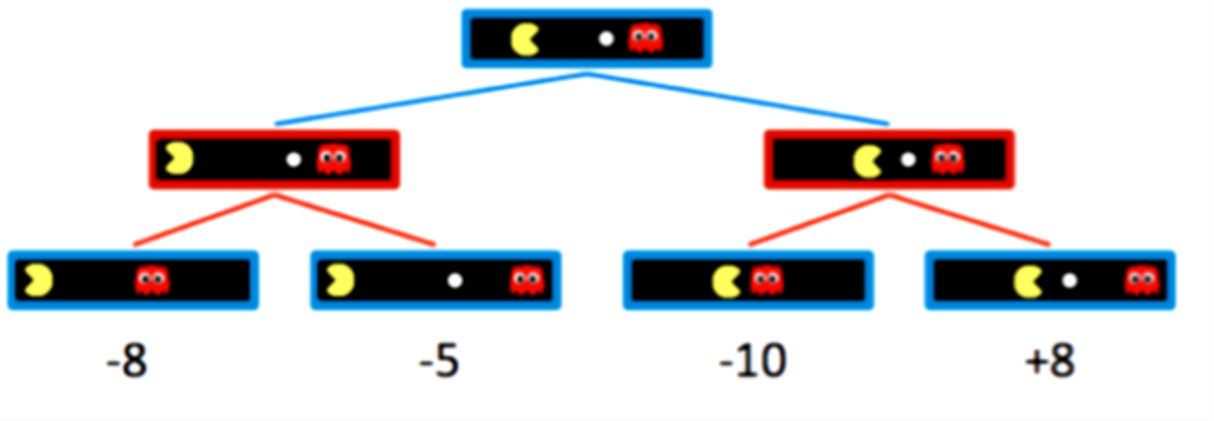
ıcman from eating the

g to a game tree where the

Adversarial Search

Minimax

A Depth-2 tree:



The root node controlled by Pacman has a value of $\max(-8, -10) = -8$

The two ghost nodes have values:

$$\begin{cases} \min(-8, -5) = -8 \\ \min(-10, +8) = -10 \end{cases}$$

Although Pacman wants the score of +8 that he can get if he ends up in the rightmost child state, through minimax he knows that an optimally-performing ghost will not allow him to have it.

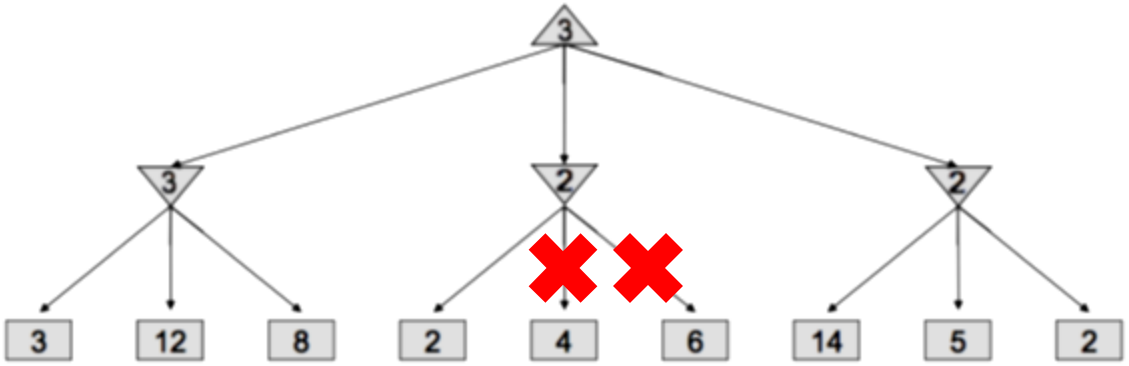
Pacman is forced to hedge his bets and counterintuitively move away from the food to minimize the magnitude of his defeat.

Adversarial Search

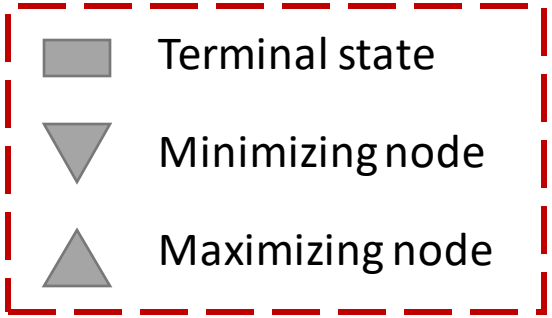
Alpha-Beta Pruning

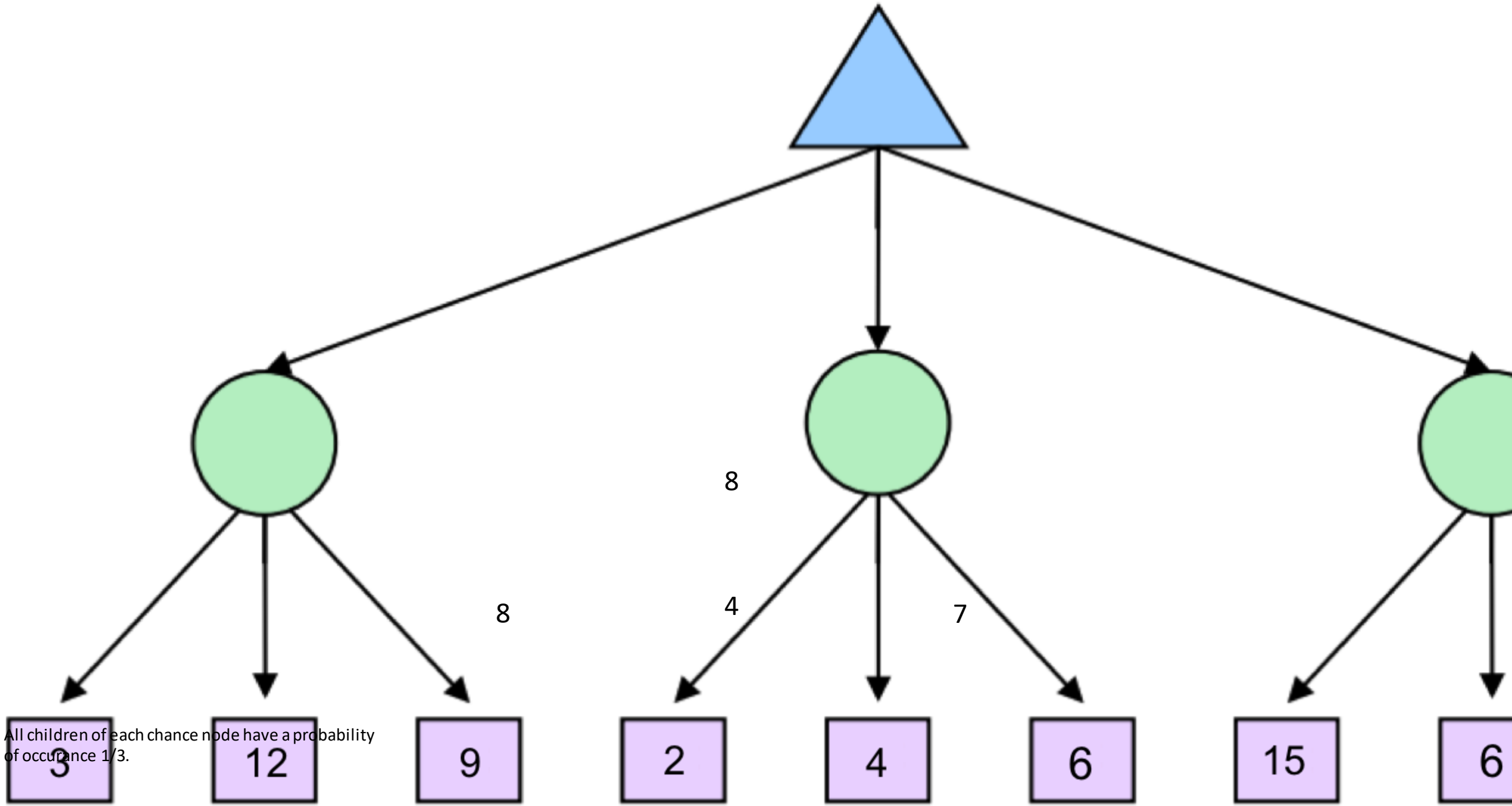
Minimax seems simple, optimal, and intuitive. But its execution is very similar to DFS and its time complexity is identical.

***α-β* pruning:** if you are trying to determine the value of a node *n* by looking at its successors, stop looking as soon as you know than *n*'s value can at best equals the optimal value of *n*'s parent.



The value of middle minimizer can be at most 2.





All children of each chance node have a probability of occurance 1/3.

Adversarial Search

Summary

We shift gears from considering standard search problems where we simply attempt to find a path from our starting point to some goal, to considering adversarial search problems.

- **Minimax**: used when our opponents behave optimally, and can be optimized using α-β pruning. Minimax provides more conservative actions than expectimax, and so tends to yield favorable results when the opponent is unknown as well.
- **Expectimax**: used when we face a suboptimal opponent, using a probability distribution over the moves we believe they will make to compute the expected values of states.

Adversarial Search

Reflex Agent (4 pts)

Minimax (5 pts)

Alpha-Beta Pruning (5 pts)

Expectimax (5 pts)

Evaluation Function (6 pts)

Project 2 Points

- The submission deadline: **March 11, 2022 23:59**
- Wrap all the codes into a ZIP file for submission in Moodle.
- The project may be done in a group of two people at most.
- This project assignment has 2 points. You should get at least 18 and 23 score from autograder to get 1 and 2 points, respectively. At least 1 point is required for passing the course.

Minimum score from autograder	Project points
18	1
23	2

- You may reach me by email: miika.malin@oulu.fi
- Tutoring is still available on Mondays 16:15 – 17:00 in Zoom. Last session is 28.2.


Thank You