

Adversarial Search

Anastasios Mantas, Vasileios Kyriakopoulos



University of Athens
November 2023



Main Aspects

Formulation

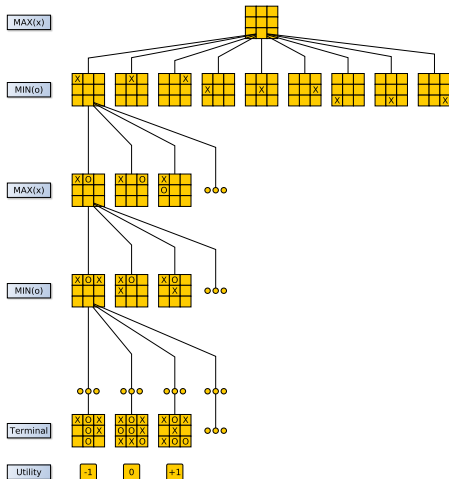
In order to perform an *Adversarial Search*, the following components are required:

- S_0 : the initial state.
- *Successor function*: yields a list of *(move, state)* pairs.
- *Terminal-test*(S): game over for S ?
- *Utility function*: Also known as *objective function*. Gives a numerical value for *terminal states*.
- *Result*(S, a): the potential estimation of performing a certain action.

Consider 2 players, *MAX* and *MIN*, assuming that *MAX* starts *first* (otherwise the inverse logic of our discussion holds).



Minimax





Pseudocode

MAX

a: MAX's best option on path to root

```
def max_value(state, a, b):  
    initialize v =  $-\infty$   
    for successor in state.successors:  
        v = max(v, value(successor, a, b))  
        if v > b:  
            return v  
        a = max(a, v)  
    return v
```



Pseudocode

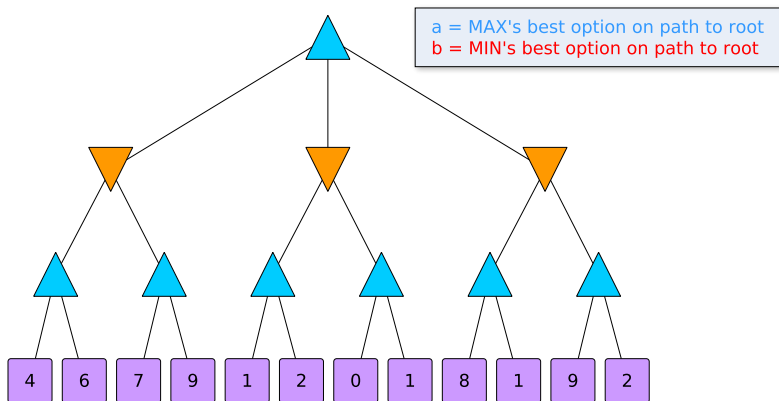
MIN

b: MIN's best option on path to root

```
def min_value(state, a, b):  
    initialize v =  $+\infty$   
    for successor in state.successors:  
        v = min(v, value(successor, a, b))  
        if v < a:  
            return v  
        b = min(b, v)  
    return v
```

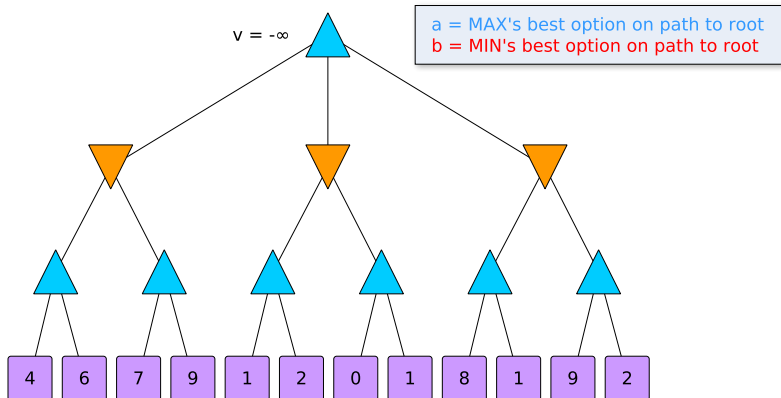


Example 1 (1/38)



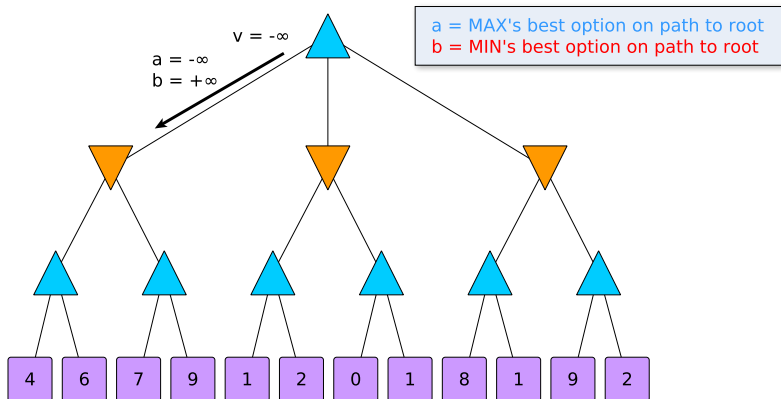


Example 1 (2/38)



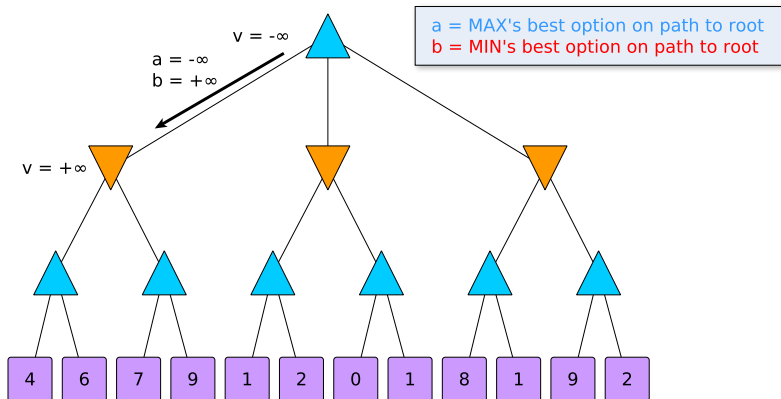


Example 1 (3/38)



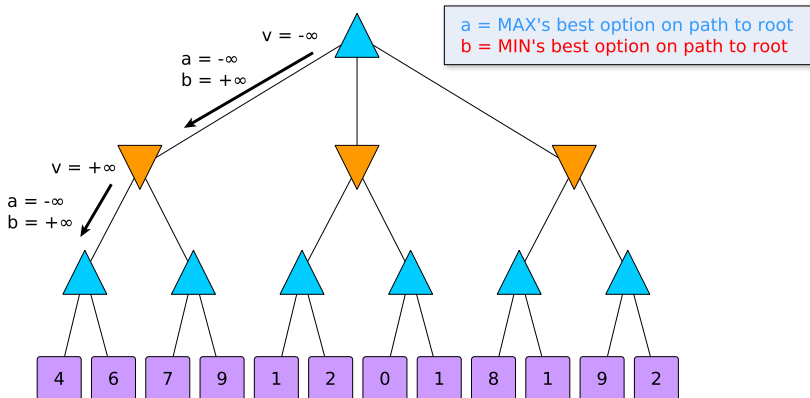


Example 1 (4/38)



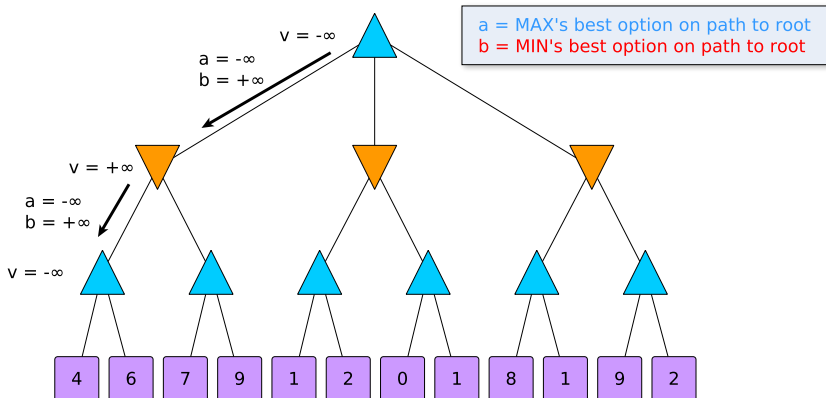


Example 1 (5/38)



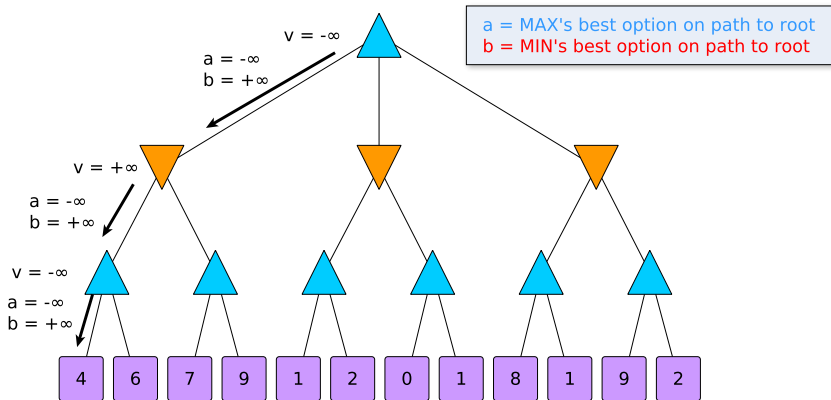


Example 1 (6/38)



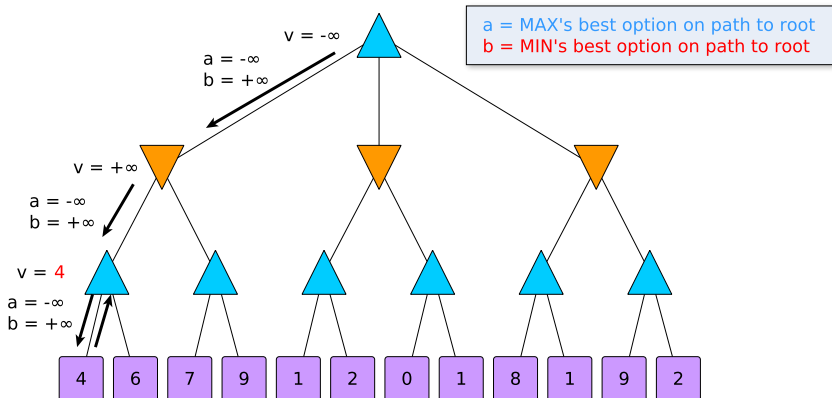


Example 1 (7/38)



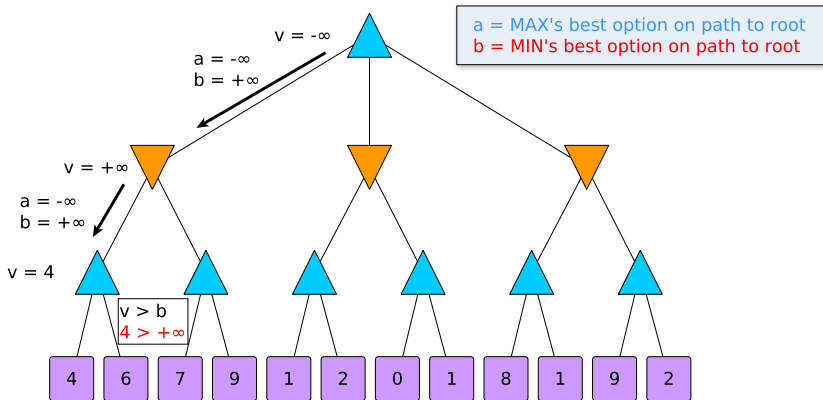


Example 1 (8/38)



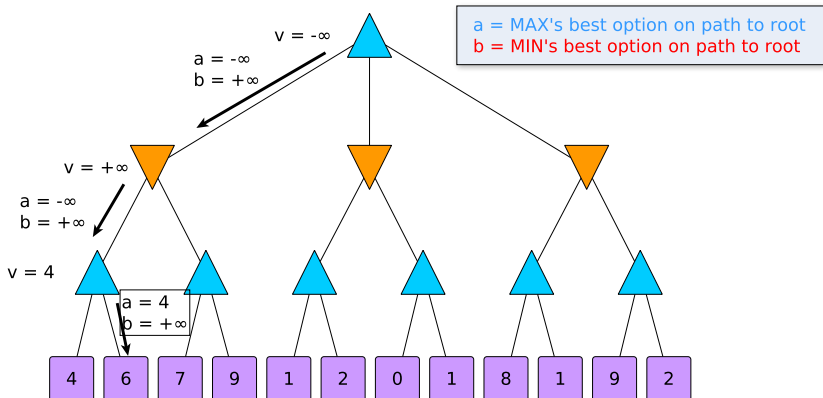


Example 1 (9/38)



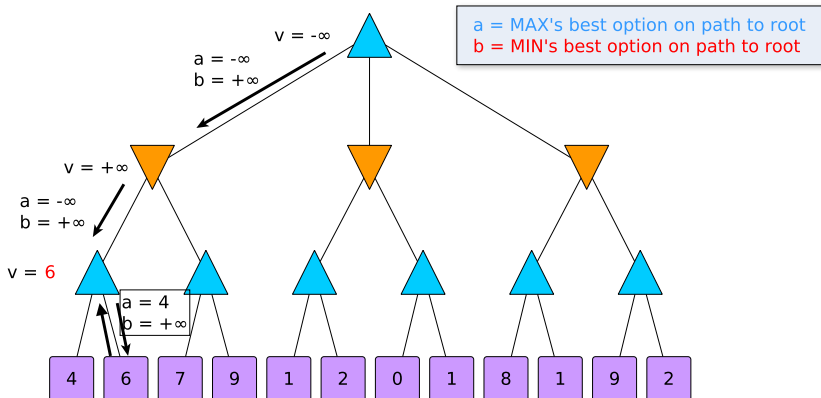


Example 1 (10/38)



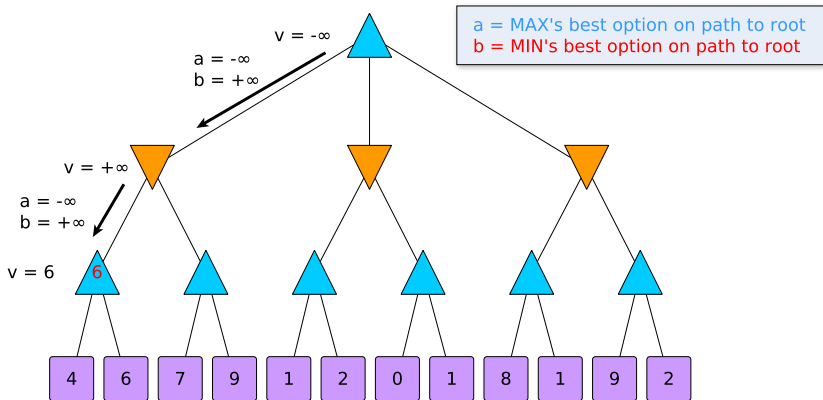


Example 1 (11/38)



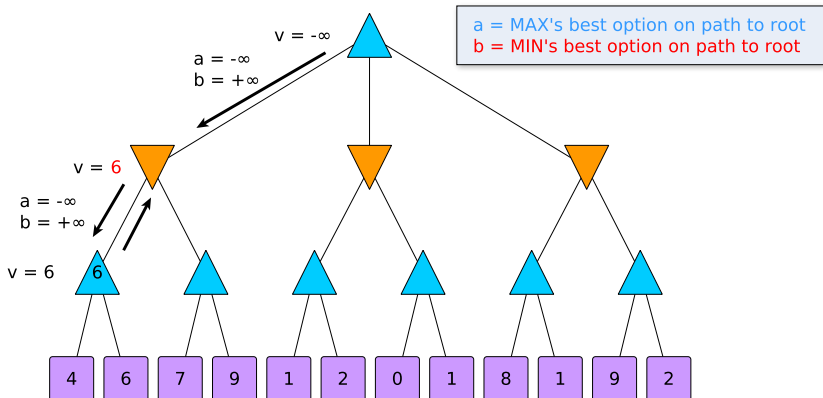


Example 1 (12/38)



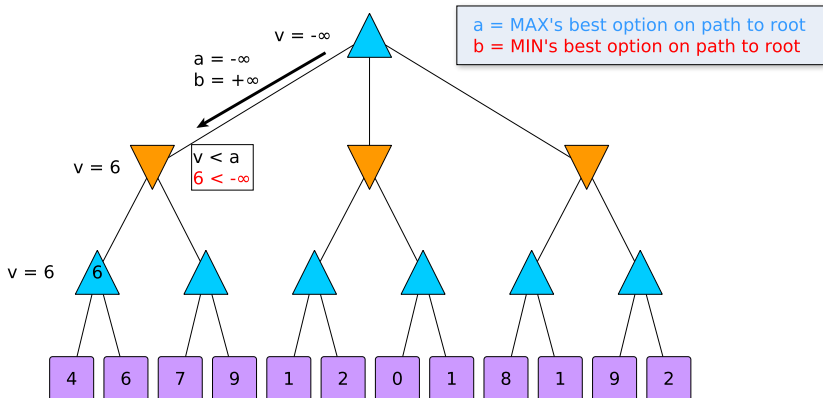


Example 1 (13/38)



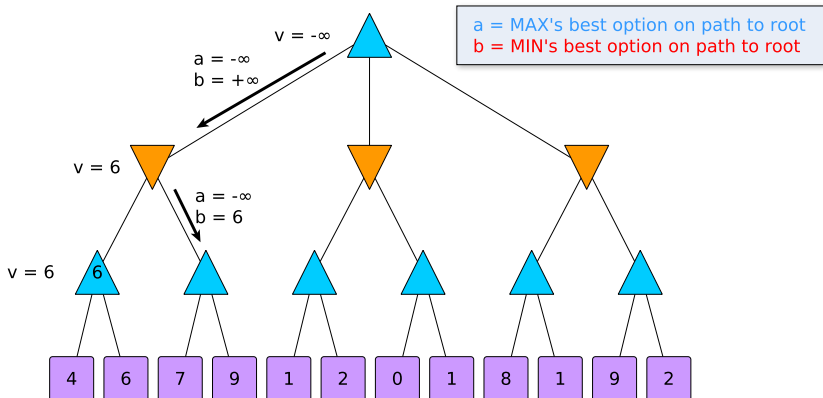


Example 1 (14/38)



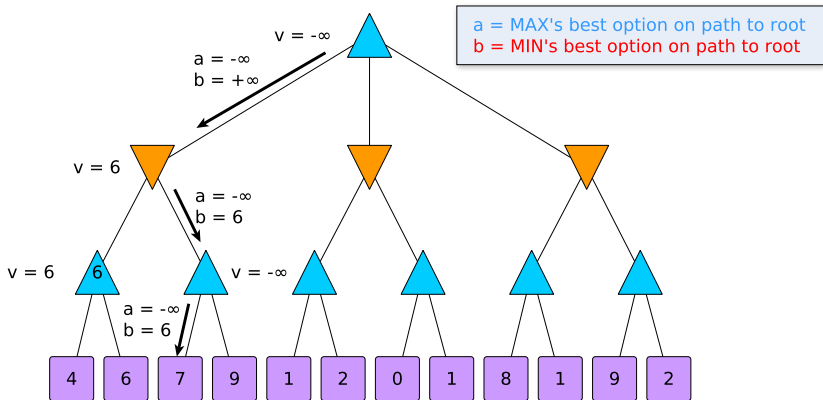


Example 1 (15/38)



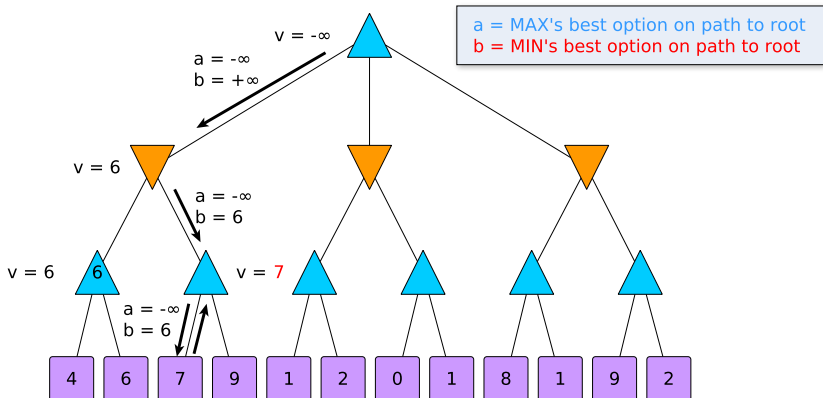


Example 1 (16/38)



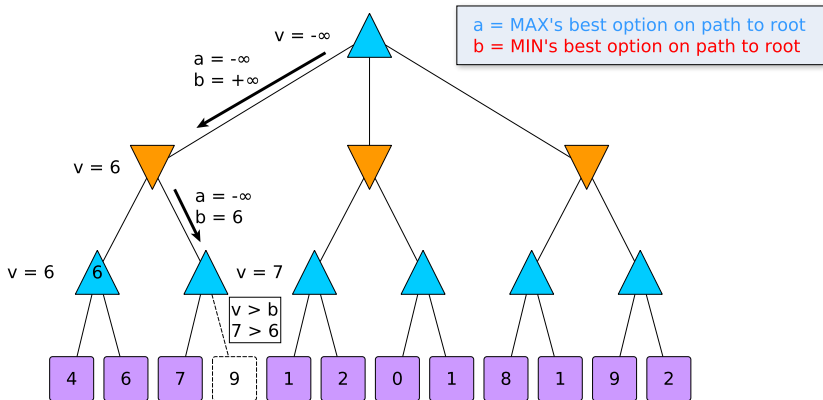


Example 1 (17/38)



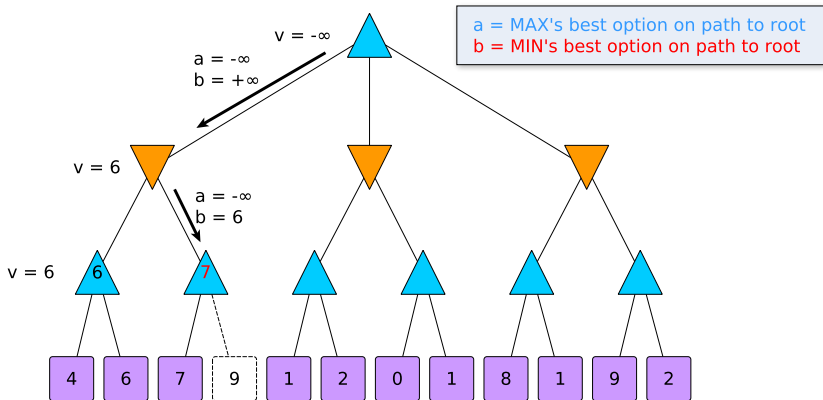


Example 1 (18/38)



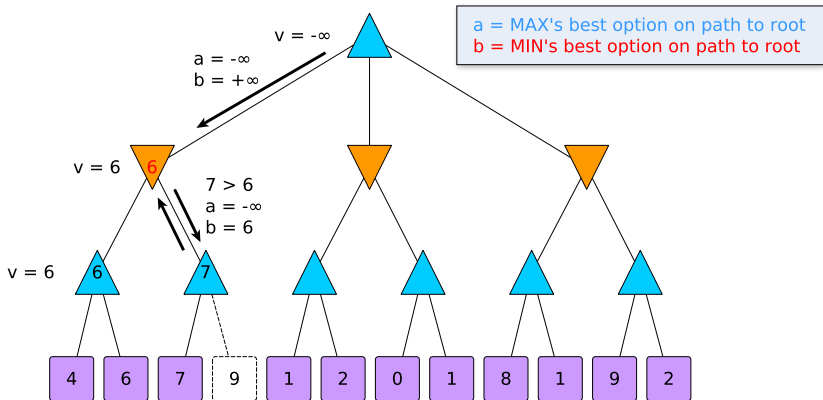


Example 1 (19/38)



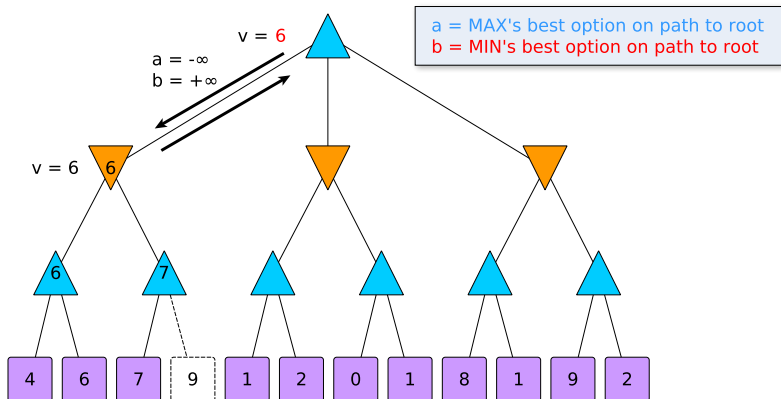


Example 1 (20/38)



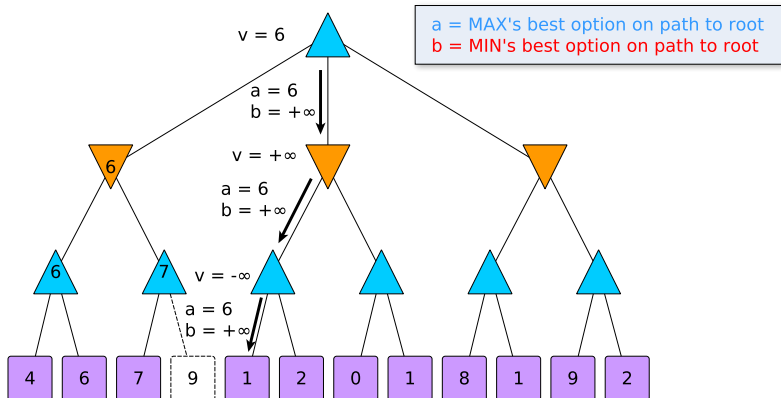


Example 1 (21/38)



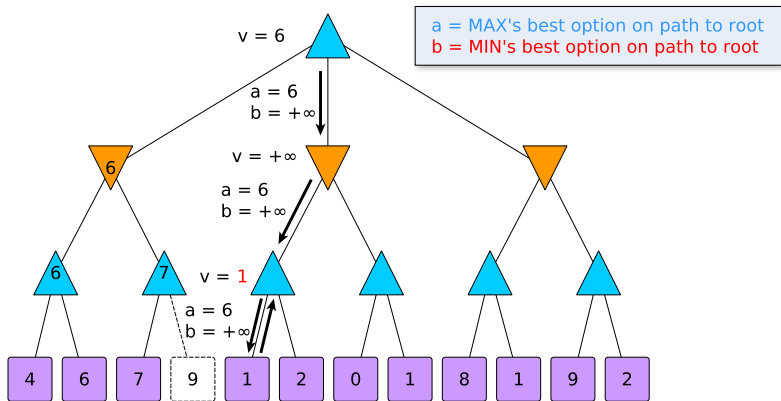


Example 1 (22/38)



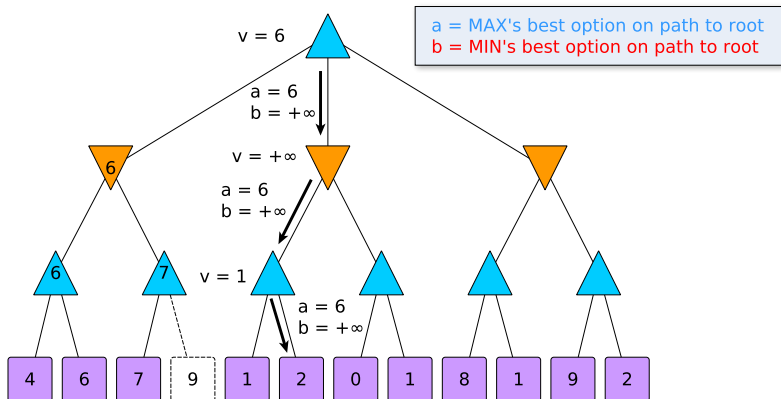


Example 1 (23/38)



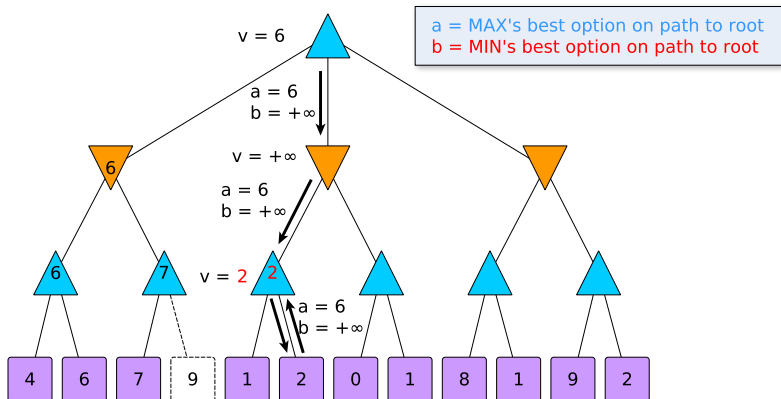


Example 1 (24/38)



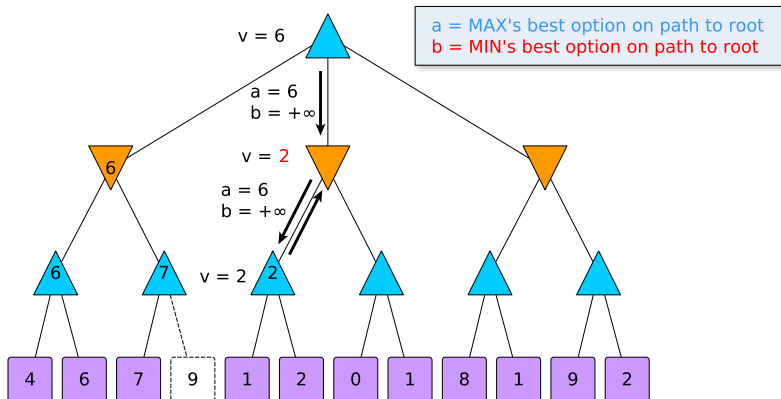


Example 1 (25/38)



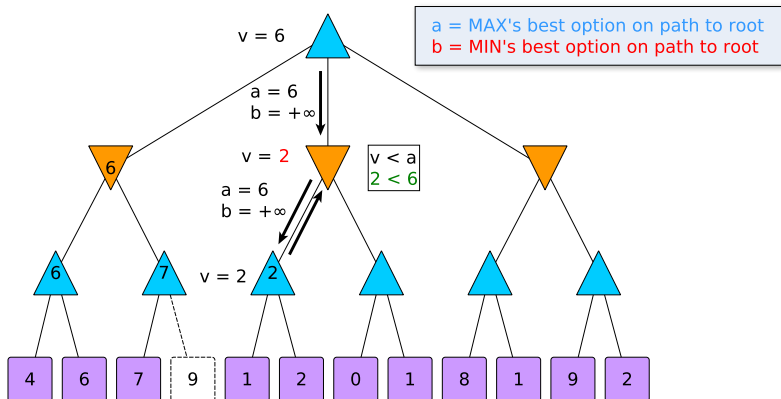


Example 1 (26/38)



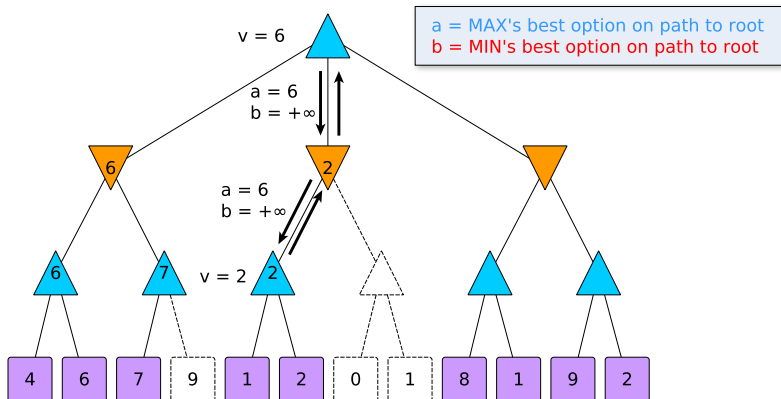


Example 1 (27/38)



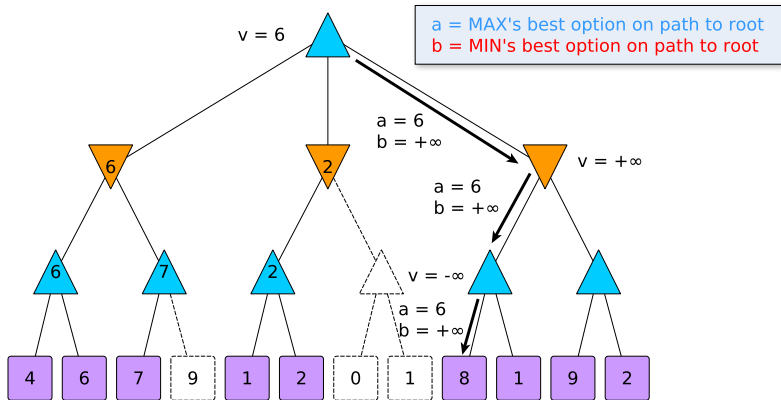


Example 1 (28/38)



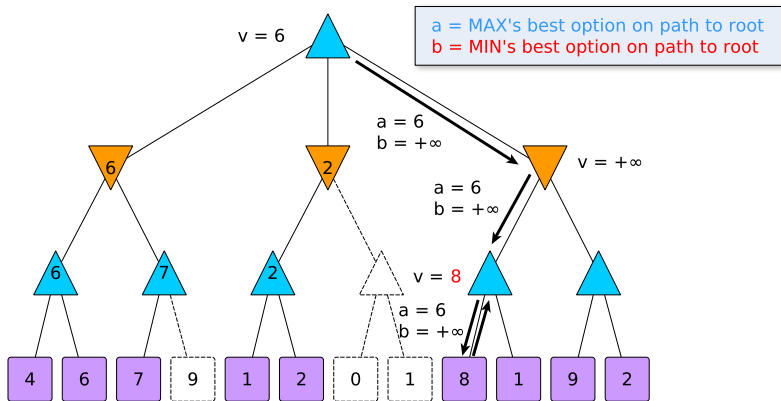


Example 1 (29/38)



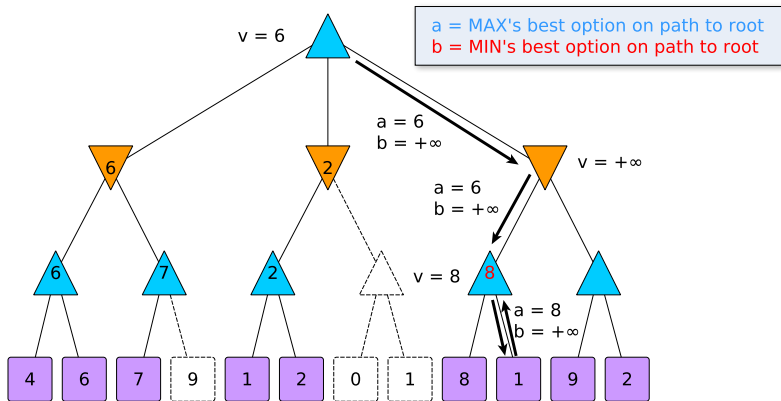


Example 1 (30/38)



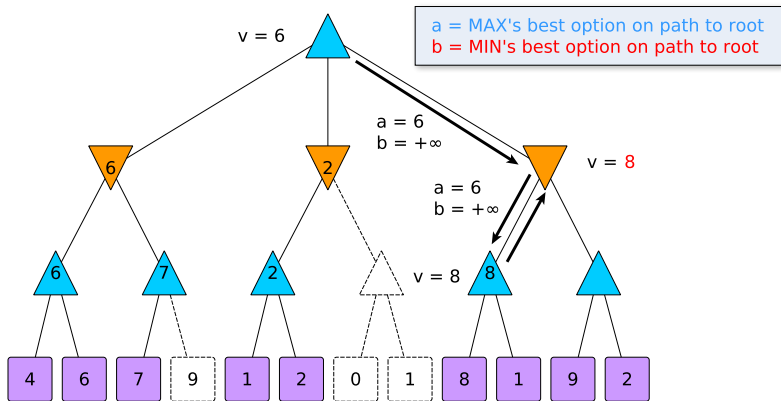


Example 1 (31/38)



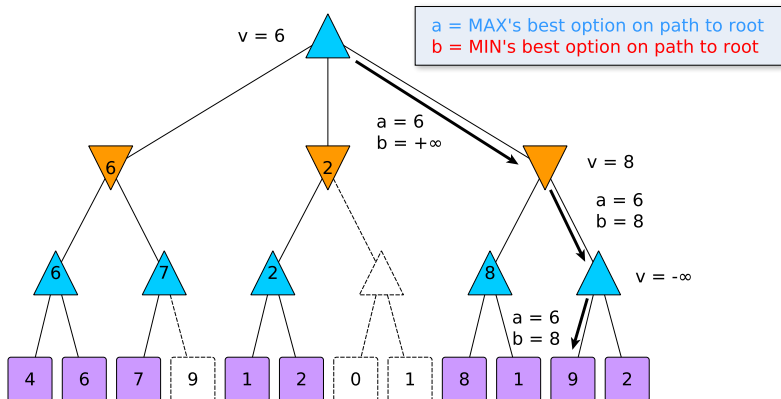


Example 1 (32/38)



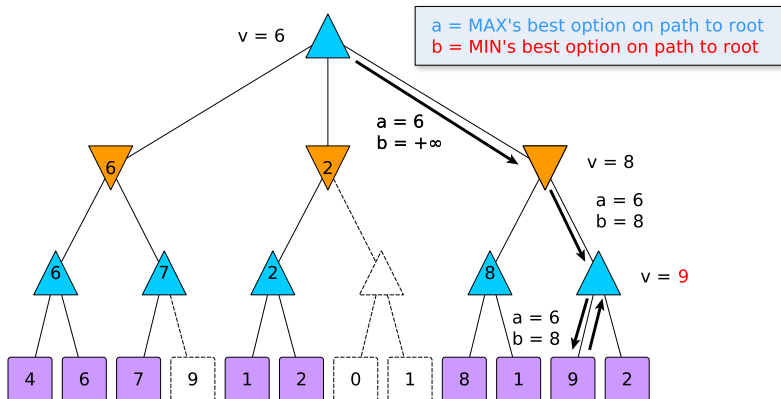


Example 1 (33/38)



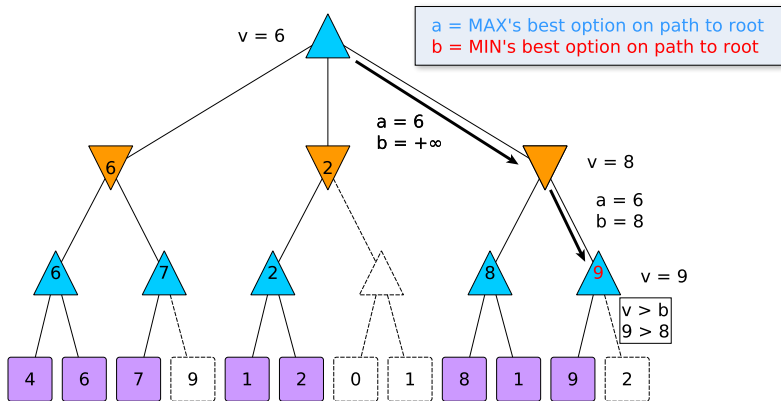


Example 1 (34/38)



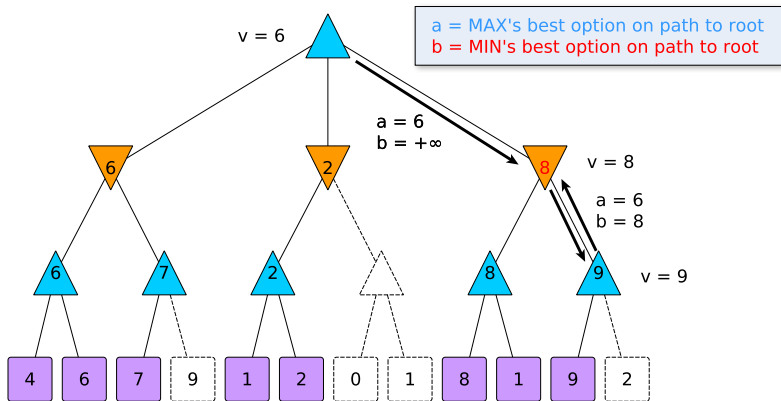


Example 1 (35/38)



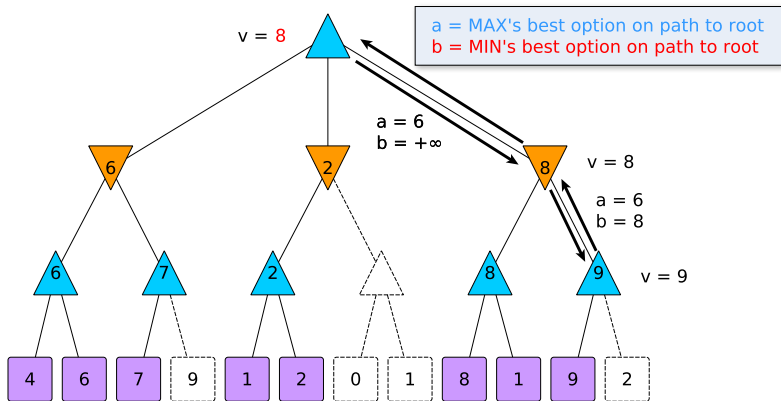


Example 1 (36/38)



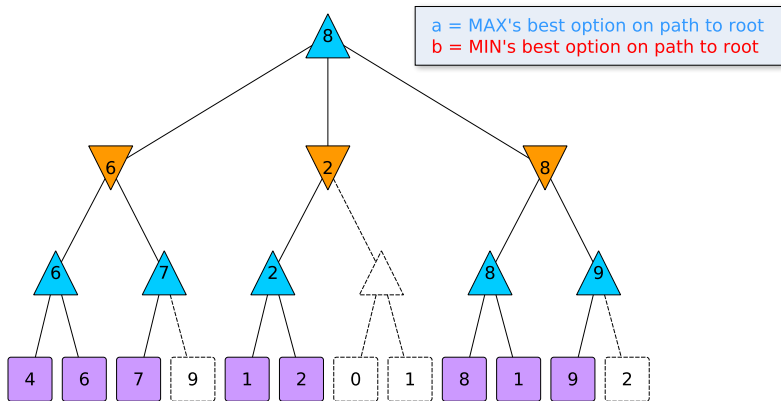


Example 1 (37/38)





Example 1 (38/38)





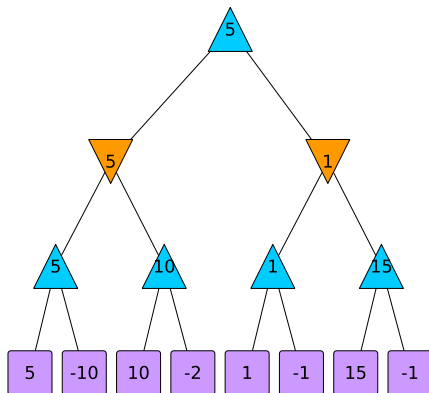
Example 2 (1/3)

Question

Given the next zero-sum game, circle **MAX**'s optimal next move on the graph and state the **minimax** value of the game. **Explain**.

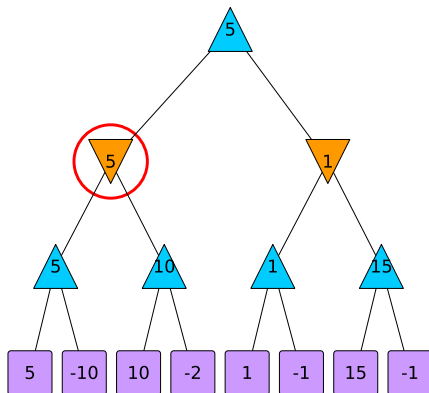


Example 2 (2/3)





Example 2 (3/3)





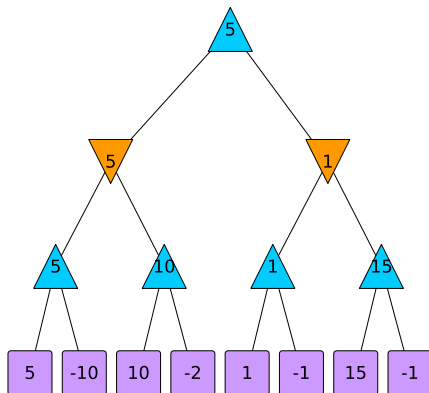
Example 3 (1/3)

Question

Same as example 2, with *a-b pruning*.

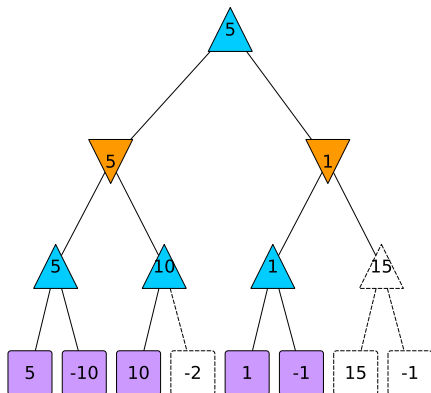


Example 3 (2/3)





Example 3 (3/3)





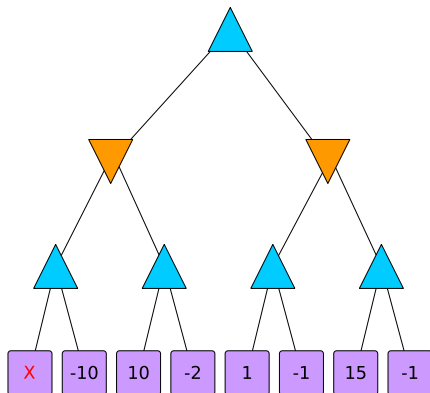
Example 4 (1/4)

Question

Modify the previous game by giving the first leaf a value of X . Find the *domain* of X in which MAX will select the left action.

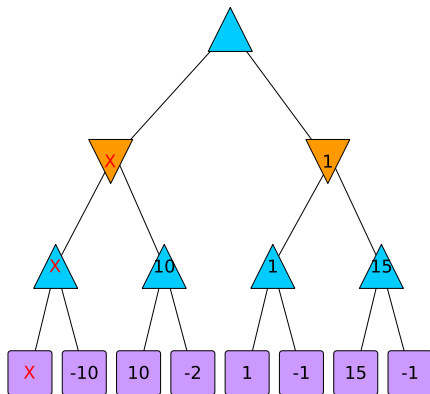


Example 4 (2/4)





Example 4 (3/4)





Example 4 (4/4)

Answer

As long as $-10 < X < 10$, **MIN** will choose the action leading to **X**. Because the **right branch** has a **value** of **1**, for any $X > 1$ (or $X \geq 1$, also acceptable), **MAX** chooses **left**.

Be careful!

- $X \leq 10$. Even if $X > 10$, we still move left initially.
- $X = 10$. **Minimax** is equal to **expectimax** but we want it to be more.



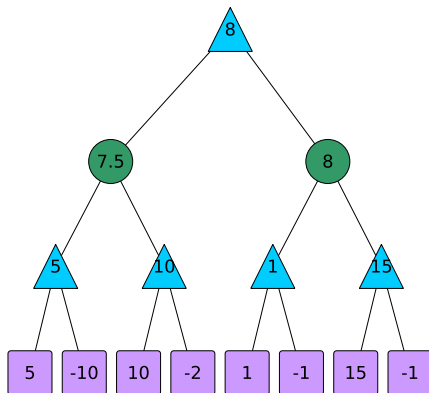
Example 1 (1/3)

Question

Same game as the previous examples, but **MIN** acts randomly in a **uniform** fashion. Circle **MAX**'s optimal move, show the **cost** value and **explain**.

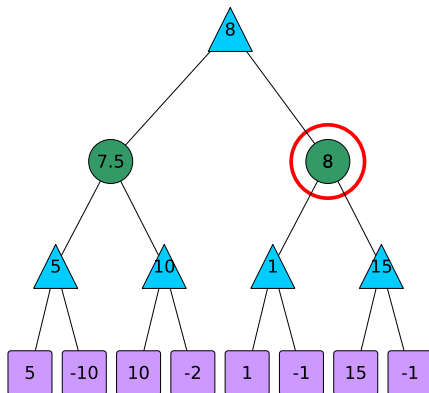


Example 1 (2/3)





Example 1 (3/3)





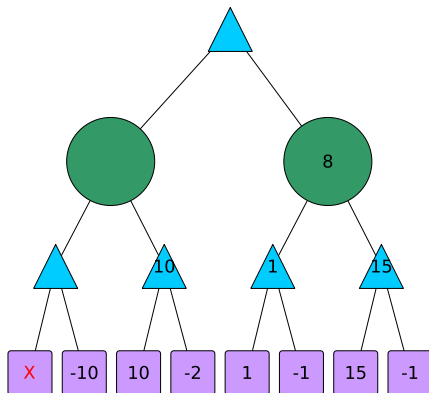
Example 2 (1/4)

Question

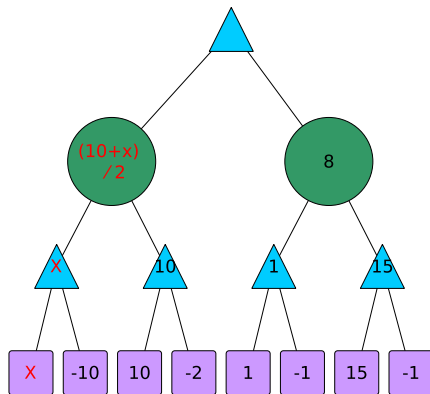
Modify the previous game by giving the first leaf a value of X . Find the *domain* of X in which MAX will select the left action.



Example 2 (2/4)



Example 2 (3/4)





Example 2 (4/4)

Answer

We need to compare 8 with $\frac{10+X}{2}$. Proper calculations show that **MAX** will select the left action if $X > 6$ (or $X \geq 6$, also acceptable).

Thank you very much for your
attention!

Original slides by: Christos Papaloukas, Iosif Angelidis