

ALGORYTM ALFA - BETA

Konsultacja merytoryczna
dr inż. Piotr Beling

Algorytm Alfa-Beta

- Algorytm przeszukujący, redukujący liczbę węzłów, które muszą być rozwiązywane w drzewach przeszukujących przez algorytm min-max
- Jest to efektywny algorytm często stosowany przy programowaniu gier logicznych. [1]
- Pozwala na szybsze znalezienie wartości mini-maxowej bez konieczności przeszukiwania całego drzewa.
- Umożliwia pomijanie pewnych fragmentów drzewa poszukiwań (odcinanie gałęzi).
- W stosunku do algorytmu mini-max ta metoda przyjmuje 2 dodatkowe parametry (α , β) które tworzą przedział obustronnie otwarty.
- Parametry te mówią o tym w jakim przedziale interesują nas wyniki. Jeżeli okaże się, że wynik jest poza przedziałem (α , β) to proces może zostać przerwany.
- Jeżeli wartość mini-maxowa jest w tym przedziale to dostajemy wartość dokładną, jeżeli nie jest w tym przedziale to otrzymujemy szacowanie z góry bądź z dołu tej wartości dokładnej.

Algorytm Alfa-Beta

- α to ograniczenie dolne dla wierzchołków Max (najwyższa wartość jaką dotychczas osiągnął gracz Max) .
- β to ograniczenie górne dla wierzchołków Min (najniższa wartość jaką dotychczas osiągnął gracz Min).
- Wartość ograniczenia α ustalana jest w wierzchołku Max , natomiast wartość ograniczenia β w wierzchołku Min .
- Odcięcie α wykonywane jest w wierzchołku Min, a odcięcie β w wierzchołku Max.
- Wtedy gdy zachodzi warunek $\alpha \geq \beta$, nie ma potrzeby analizowania dalszych następników danego stanu.
- Przedział (α, β) jest zawsze obustronnie otwarty.

Algorytm Alfa-Beta

- Dwóch przeciwników (min-max) posiadają pełną informację o stanie gry i wszystkich możliwych ruchach
- Gracz Max, ponieważ:
 - maksymalizuje rezultat końcowy
 - każdy wzrost wartości oznacza poprawę dla tego gracza i równoważną stratę dla przeciwnika
- Gracz Min, ponieważ:
 - minimalizuje rezultat końcowy
 - każdy spadek wartości oznacza poprawę dla tego gracza i równoważną stratę dla przeciwnika
- Min-Max wymaga przemiennej wywołań rekurencyjnych dwóch graczy (raz dla gracza MAX, dwa dla gracza MIN, itd.) [2]

Cechy Algorytmu Alfa-Beta

- Odcięcie w algorytmie alfa – beta jest zawsze korzystne, ponieważ otrzymujemy w rezultacie lepszą złożoność czasową.
- Efektywność algorytmu zależy w wielkim stopniu od kolejności następników i występowania odcięć.
- Idealna sytuacja zachodzi wtedy, kiedy odcięcie występuje jak najszybciej tzn. najlepiej zaraz po sprawdzeniu pierwszego następnika.

Złożoność algorytmu Alfa-Beta

Dla danej głębokości (d) i stałego stopnia rozgałęzienia(b)

- Najlepszy przypadek : $O\left(b^{\frac{d}{2}}\right)$
- Najgorszy przypadek : $O(b^d)$ - brak odcięć (tak jak w MinMax)
- Średni przypadek : $O(b^{0,75d})$ [3]

Funkcja heurystyczna

Funkcja heurystyczna – metoda służąca do znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego bądź też prawidłowego. Rozwiązanie te stosuje się najczęściej gdy dany algorytm jest zbyt kosztowny. Metoda ta służy również do tego aby znaleźć rozwiązanie przybliżone, na podstawie których wylicza się ostateczne rezultaty pełnym algorytmem. Dzięki temu możemy zmniejszyć czas działania programu w typowym przypadku bez poświęcania jakości rozwiązania. [4]

Funkcja heurystyczna

Efektywność algorytmu w dużym stopniu zależy od jakości używanej funkcji heurystycznej, która powinna przypisywać ocenę:

- dodatnią dla stanu, którego większą szanse wygranej posiada gracz.
- ujemną dla stanu, którego większą szansę wygranej posiada przeciwnik
- tym większą co do wartości bezwzględnej im przewaga szans wygranej gracza albo przeciwnika jest większa [5]

Rozszerzenia algorytmu Alfa-Beta

- Ulepszanie funkcji oceny stanu (funkcji heurystycznej)
- Modyfikacje sposobu przeszukiwania grafu
 - Zastosowanie pamięci (np. tablica przejść)
 - Manipulowanie zakresem $\alpha - \beta$
 - Porządkowanie następników
 - Zmienna głębokość przeszukiwania
 - Przeszukiwanie eksploracyjne
- Rozwiązania sprzętowe (np. obliczenia równoległe)

Zalety algorytmu Alfa - Beta

- Umożliwia odcięcie gałęzi drzewa wyszukiwania.
- Ogranicza czas obliczenia i wyszukiwania w porównaniu do podstawowego algorytmu minimax.
- Zapobiega wykorzystywaniu dodatkowego czasu obliczeniowego i w rezultacie proces ten będzie szybszy.

Wady algorytmu Alfa - Beta

- Nie rozwiązuje wszystkich problemów tak jak algorytm minimax.
- Oceny węzła zwykle są nie dokładne lecz przybliżone i w wyniku otrzymujemy szacowania wartości danej pozycji.
- Wymaga określonego limitu odległości, gdyż w większość przypadków przeszukiwanie całego drzewa gry nie jest możliwe. Przykładem może być gra Go, którego współczynnik rozgałęzienia wynosi 360. [6]

Wariant fail-soft algorytmu Alfa-Beta

- W klasycznej wersji zawsze wartością zwracaną jest wartość z przedziału (α, β)
- Wariant fail-soft zwraca dowolne wartości niezależnie od początkowego zakresu (α, β)
- Fail-soft algorytmu AlfaBeta stanowi podstawę do wszelkich jego modyfikacji wykorzystujących manipulację zakresem (α, β) . [7]

Pseudokod algorytmu Fail-Soft Alfa-Beta

```
1  int alphabeta(int depth, int alpha, int beta)
2  {
3      move bestmove;
4      int current = -WIN;
5      if (game over or depth <= 0) return winning score or eval();
6      for (each possible move m) {
7          make move m;
8          score = -alphabeta(depth - 1, -beta, -alpha)
9          unmake move m;
10         if (score >= current) {
11             current = score;
12             bestmove = m;
13             if (score >= alpha) alpha = score;
14             if (score >= beta) break;
15         }
16     }
17     return current;
18 }
```

Pseudokod algorytmu Fail-Soft Alfa-Beta

Kod zwraca wartość z przedziału alfa-beta jeżeli wynik znajdzie się w tym przedziale. W innym wypadku dostaniemy szacowanie z góry bądź z dołu wartości dokładnej.

Bieżący wynik jest przechowywany w zmiennej alfa (ograniczenie dolne), gdzie znajduje się najmniejsza możliwa wartość z przedziału α , β . Wartość zawsze będzie wzrastać od tego miejsca.

Jednym z najprostszych ulepszeń alfa-beta jest przechowywanie obecnego wyniku (current) i alfa w osobnych zmiennych.

Pseudokod używa stałej WIN do oznaczenia maksymalnego wyniku, który może zostać zwrócony przez każde wywołanie wyszukiwania alfa-beta.

Twierdzenie o wynikach algorytmu α, β

Niech $(\mathbb{S}, s, N, ev)$ – gra, $s \in \mathbb{S}$ – stan gry, $0 \leq \alpha < \beta \leq 1$.

Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz

$m = \text{MinMax}(s)$. [8]

Możemy wyróżnić następujące sytuacje:

$$\alpha \geq ab \Rightarrow ab \geq m,$$

$$\alpha < ab < \beta \Rightarrow ab = m,$$

$$\beta \leq ab \Rightarrow ab \leq m$$



Twierdzenie o wynikach algorytmu α, β

Niech $(\mathbb{S}, s, N, ev)$ – gra, $s \in \mathbb{S}$ – stan gry, $0 \leq \alpha < \beta \leq 1$.

Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz

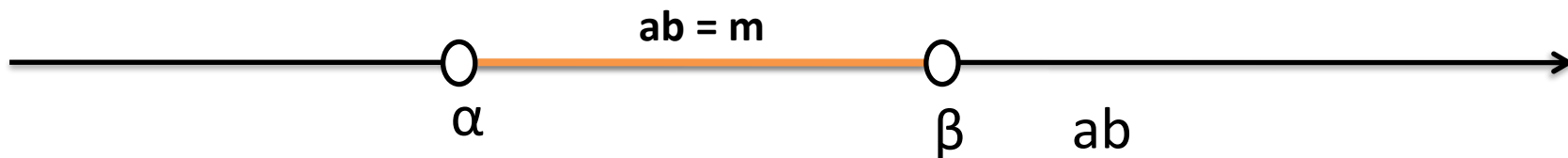
$m = \text{MinMax}(s)$. [8]

Możemy wyróżnić następujące sytuacje:

$$\alpha \geq ab \Rightarrow ab \geq m,$$

$$\alpha < ab < \beta \Rightarrow ab = m,$$

$$\beta \leq ab \Rightarrow ab \leq m$$



Twierdzenie o wynikach algorytmu α, β

Niech $(\mathbb{S}, s, N, ev)$ – gra, $s \in \mathbb{S}$ – stan gry, $0 \leq \alpha < \beta \leq 1$.

Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz

$m = \text{MinMax}(s)$. [8]

Możemy wyróżnić następujące sytuacje:

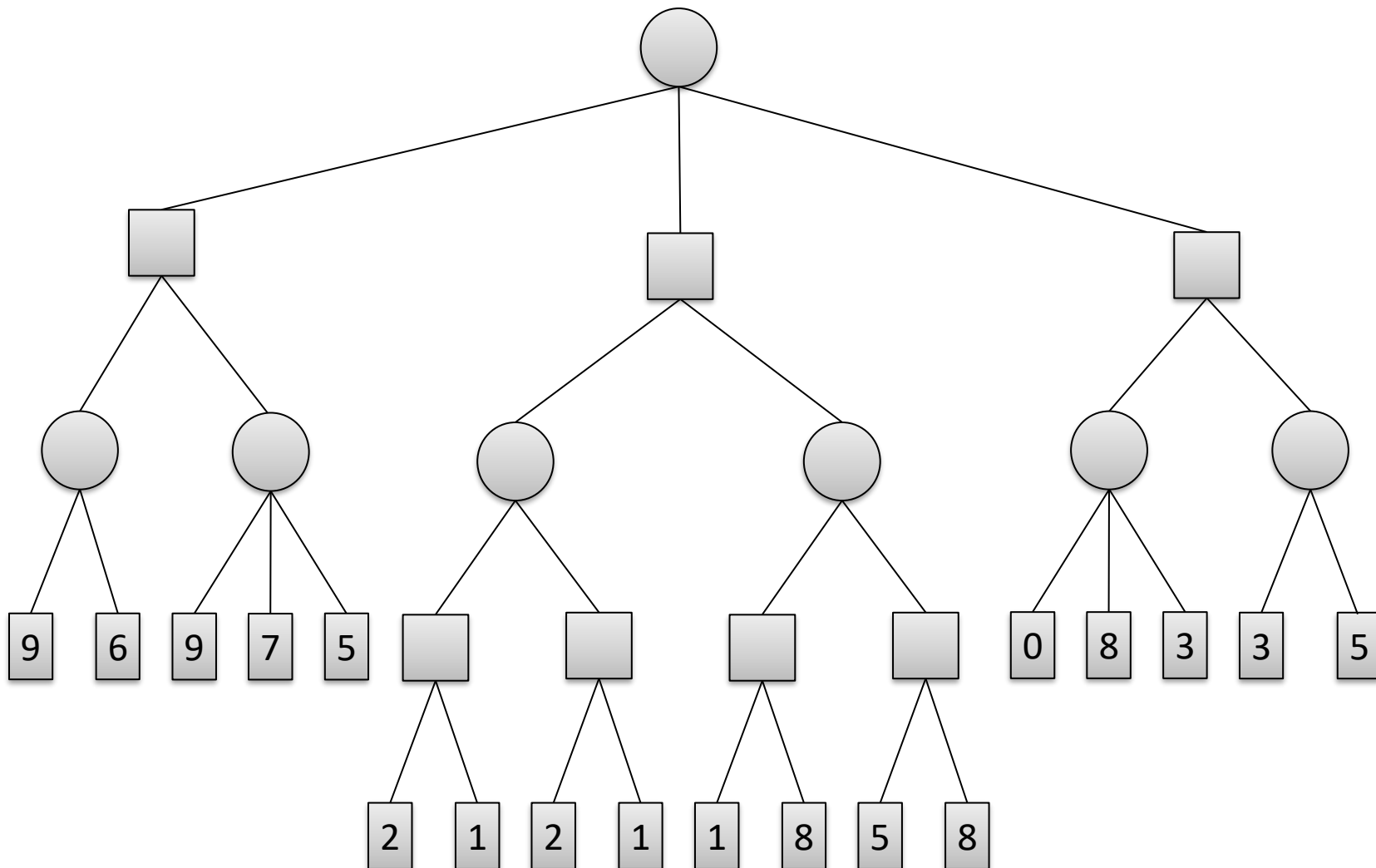
$$\alpha \geq ab \Rightarrow ab \geq m,$$

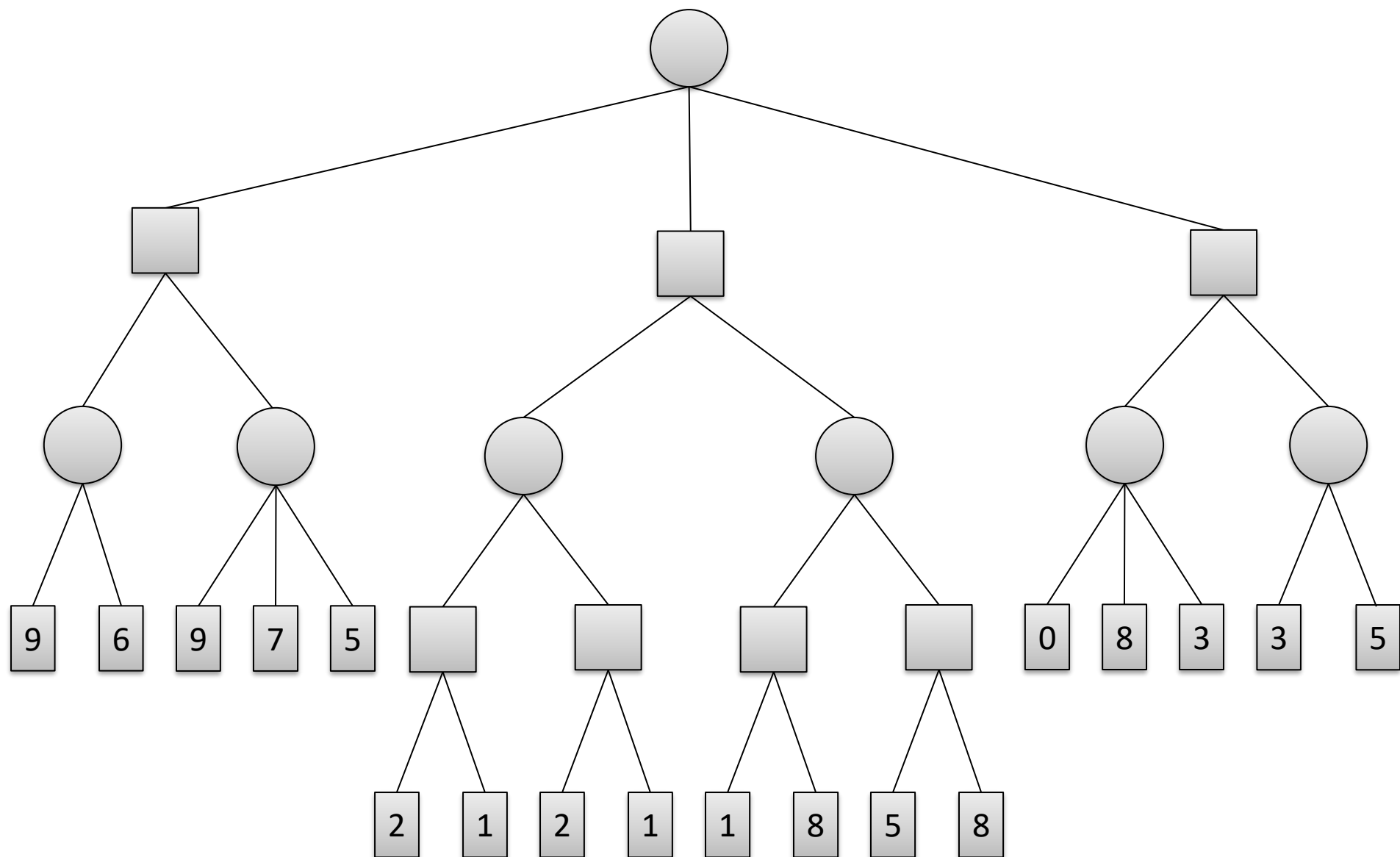
$$\alpha < ab < \beta \Rightarrow ab = m,$$

$$\beta \leq ab \Rightarrow ab \leq m$$



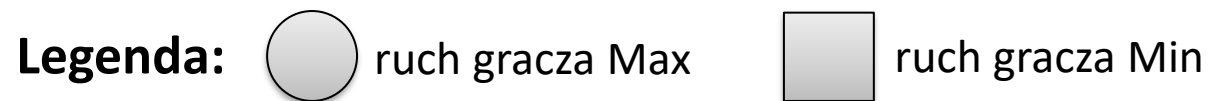
Przykład 1. Rozważmy działanie wywołanego algorytmu α - β w oknie (3,6) poniższego grafu gry:

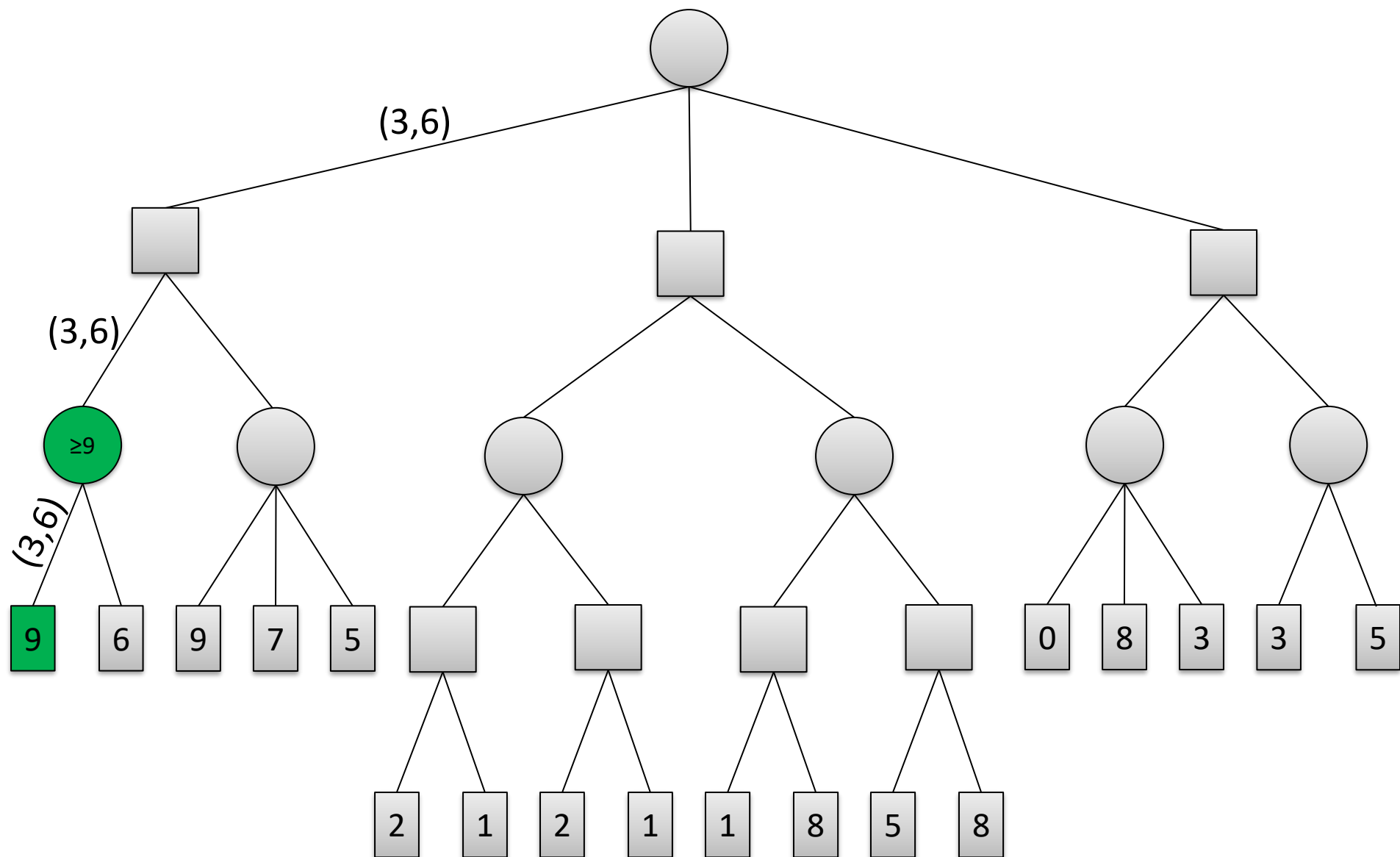




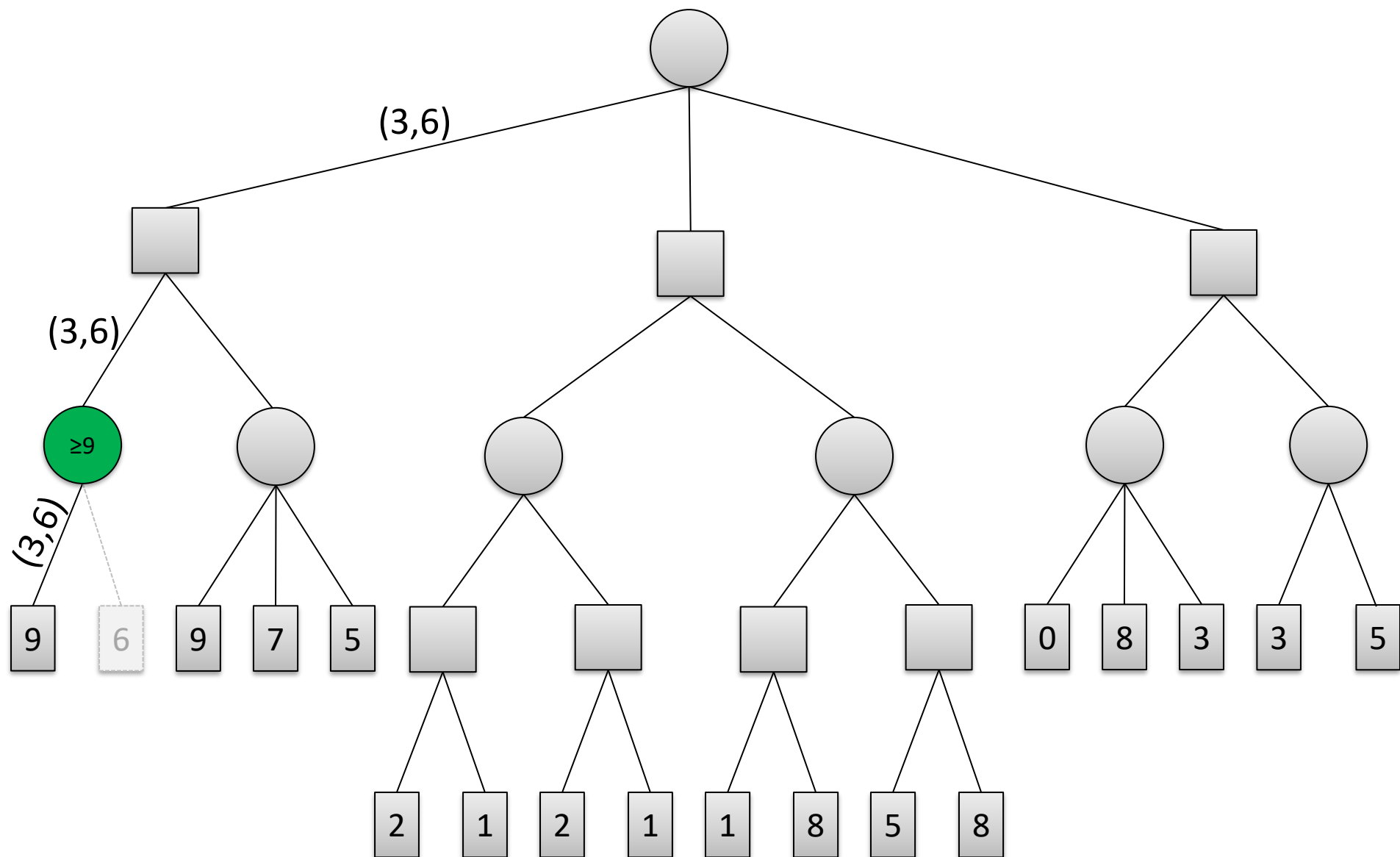
Legenda:  ruch gracza Max

 ruch gracza Min



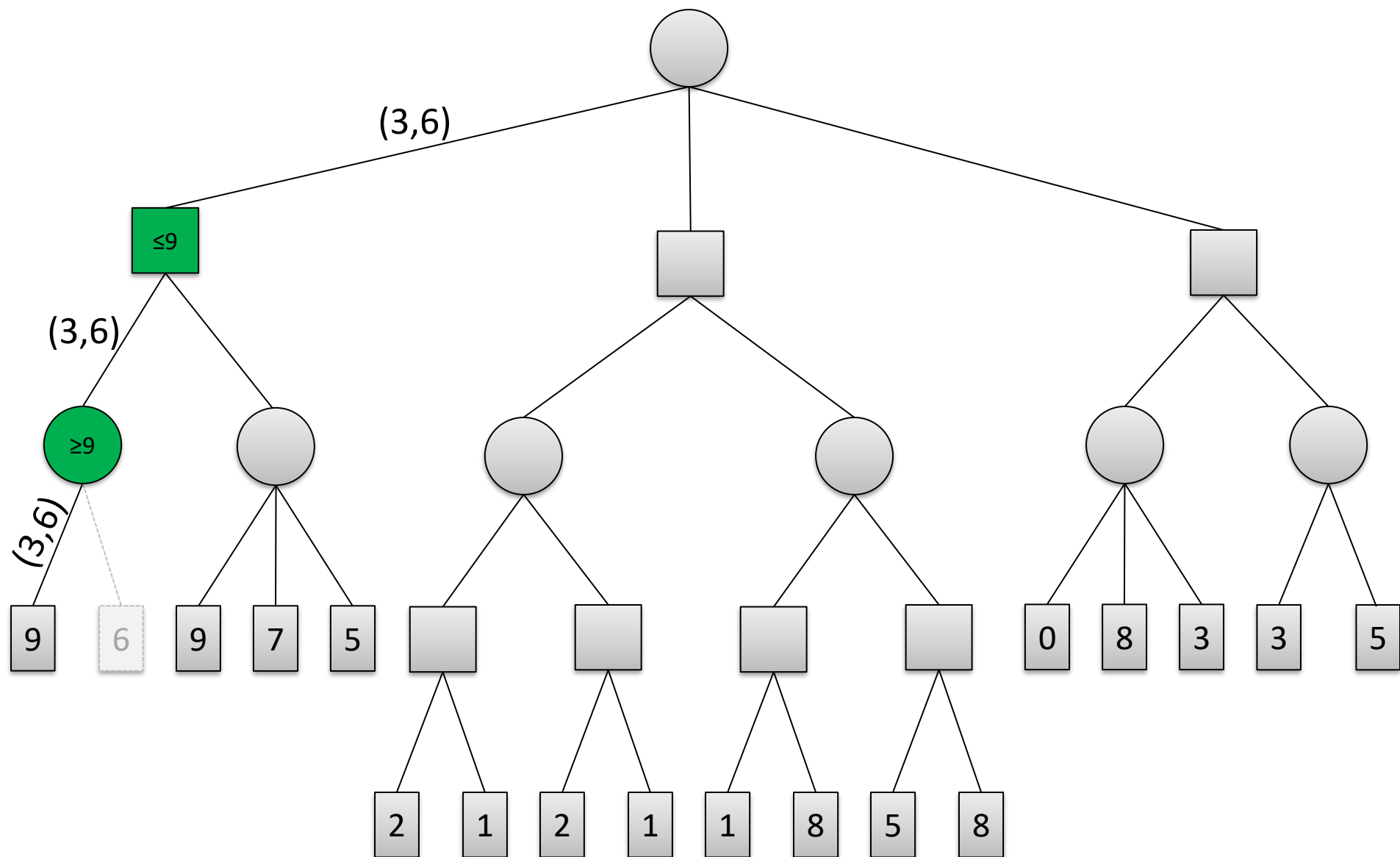


Legenda:  ruch gracza Max  ruch gracza Min



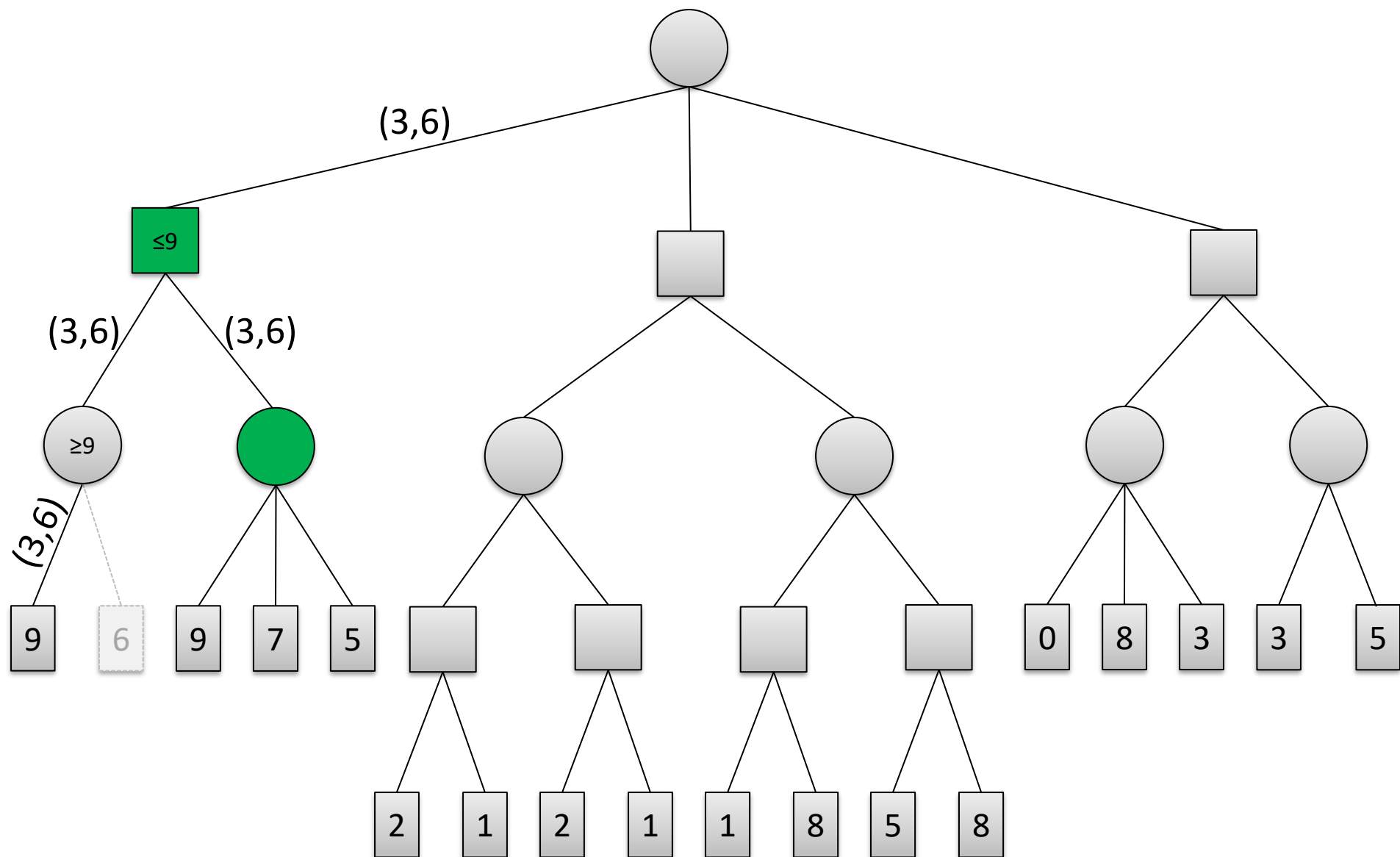
Legenda:  ruch gracza Max

 ruch gracza Min



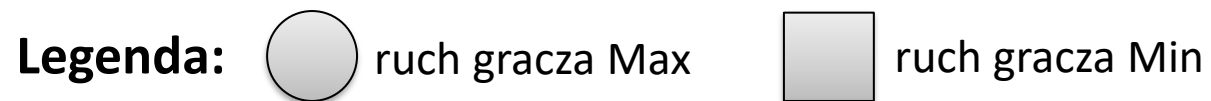
Legenda:  ruch gracza Max

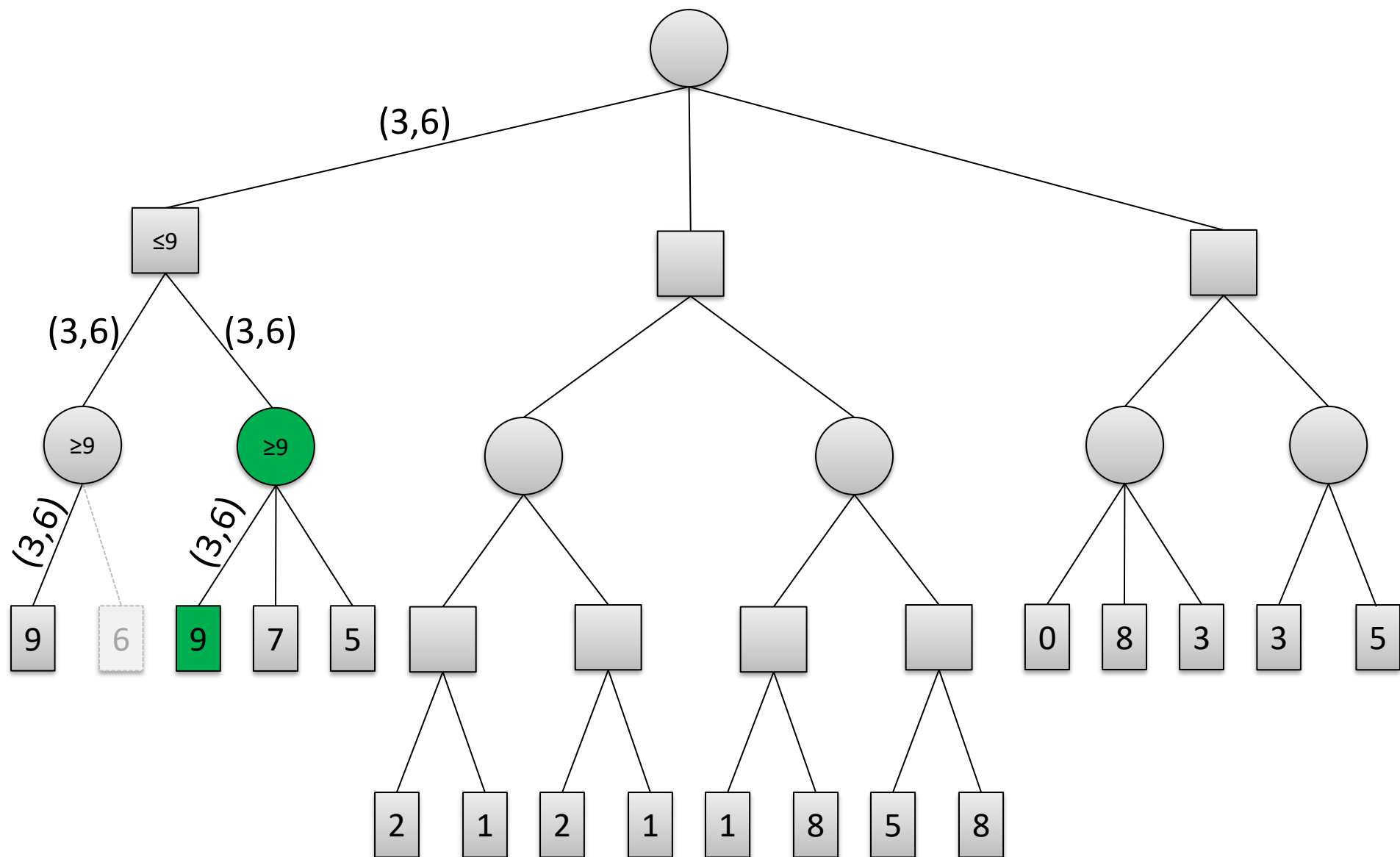
 ruch gracza Min



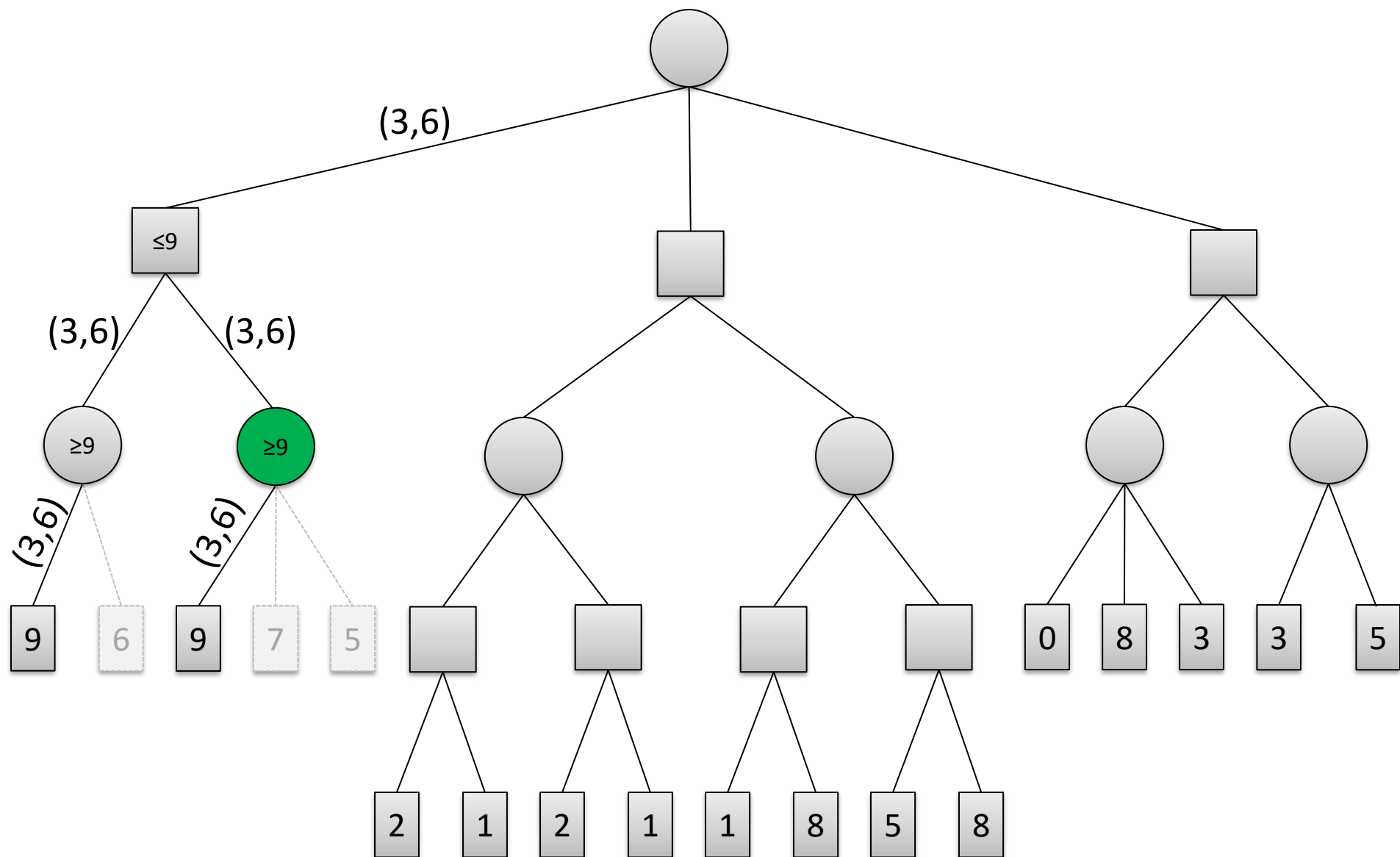
Legenda:  ruch gracza Max

 ruch gracza Min

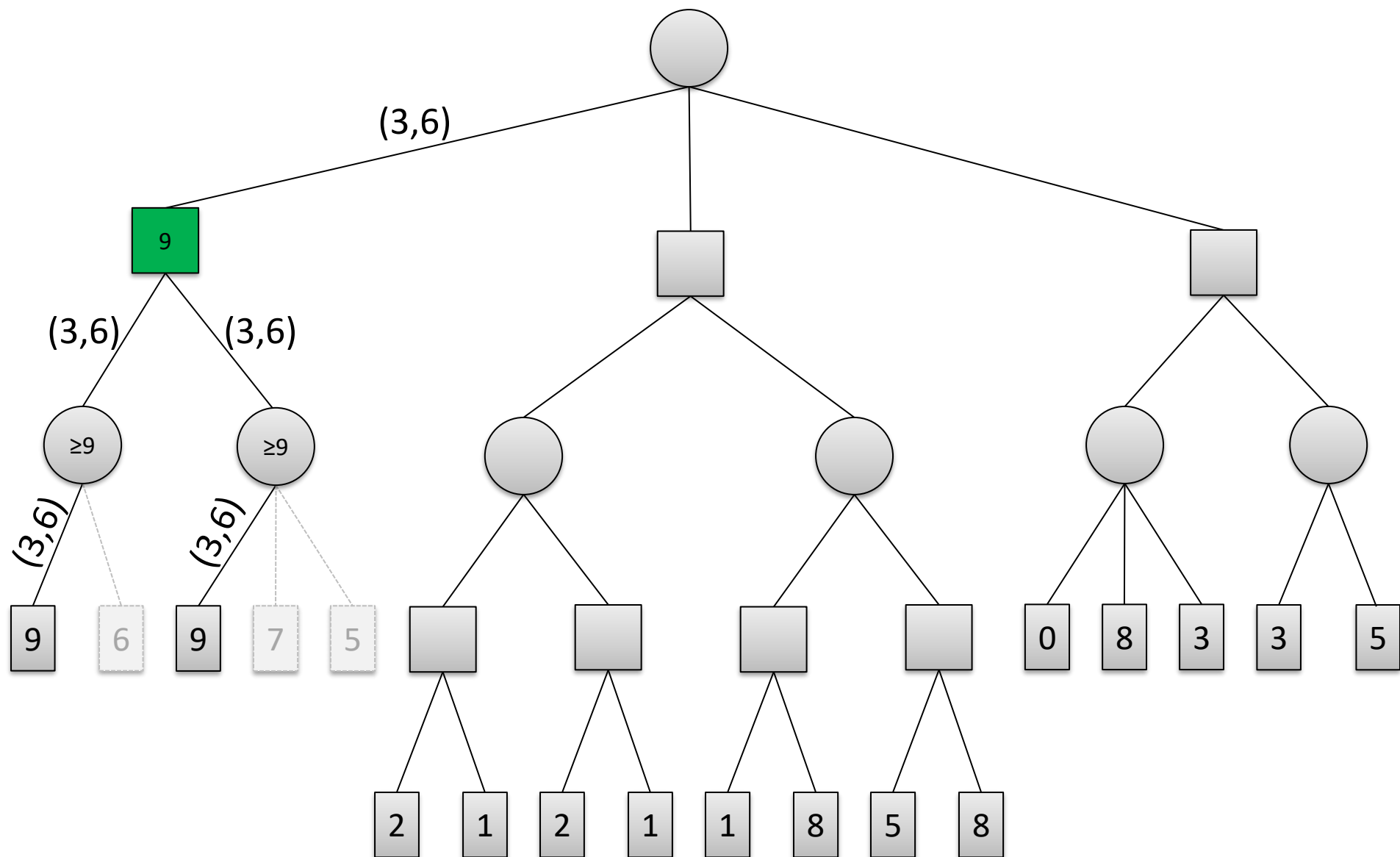




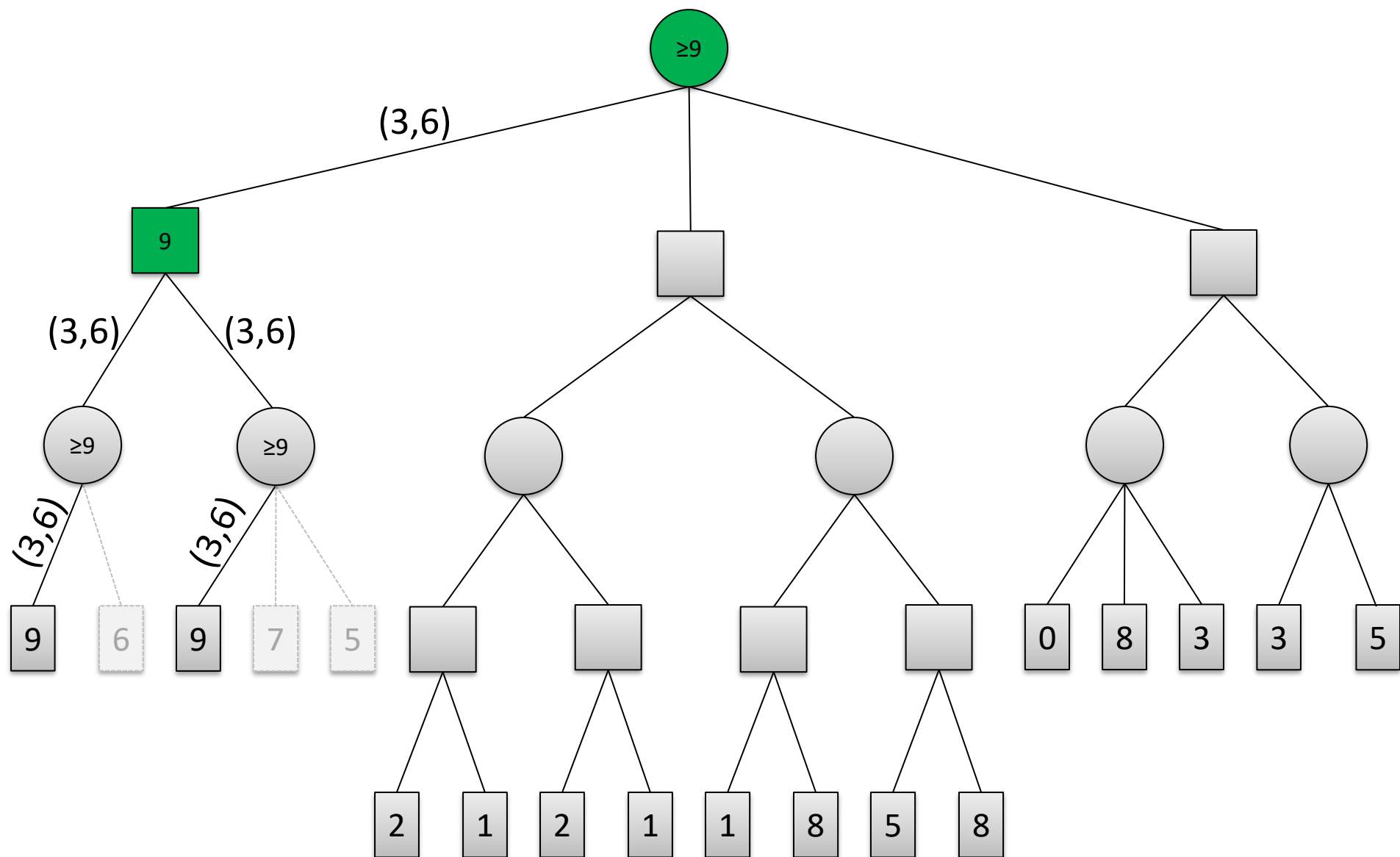
Legenda:  ruch gracza Max  ruch gracza Min



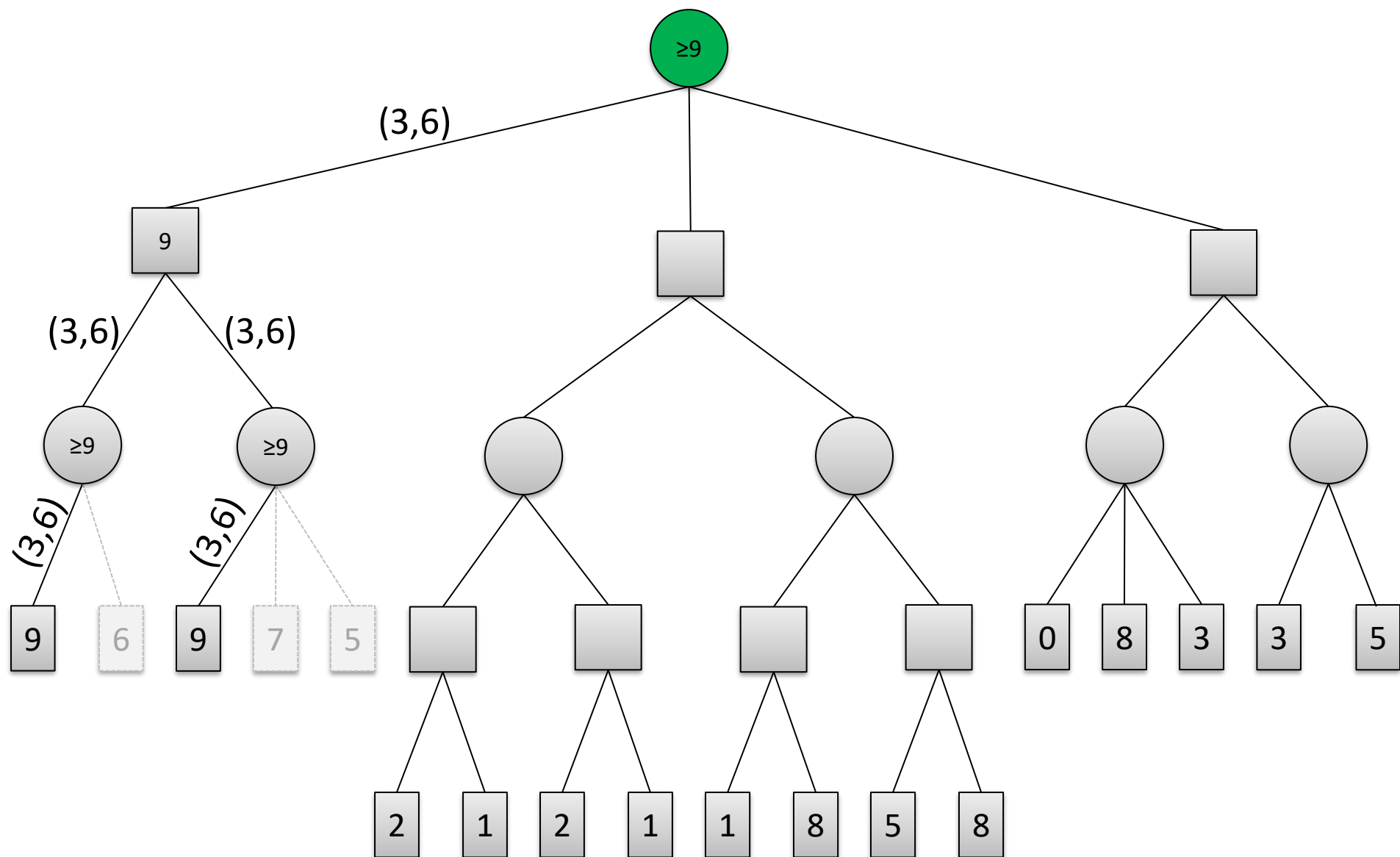
Legenda:  ruch gracza Max  ruch gracza Min

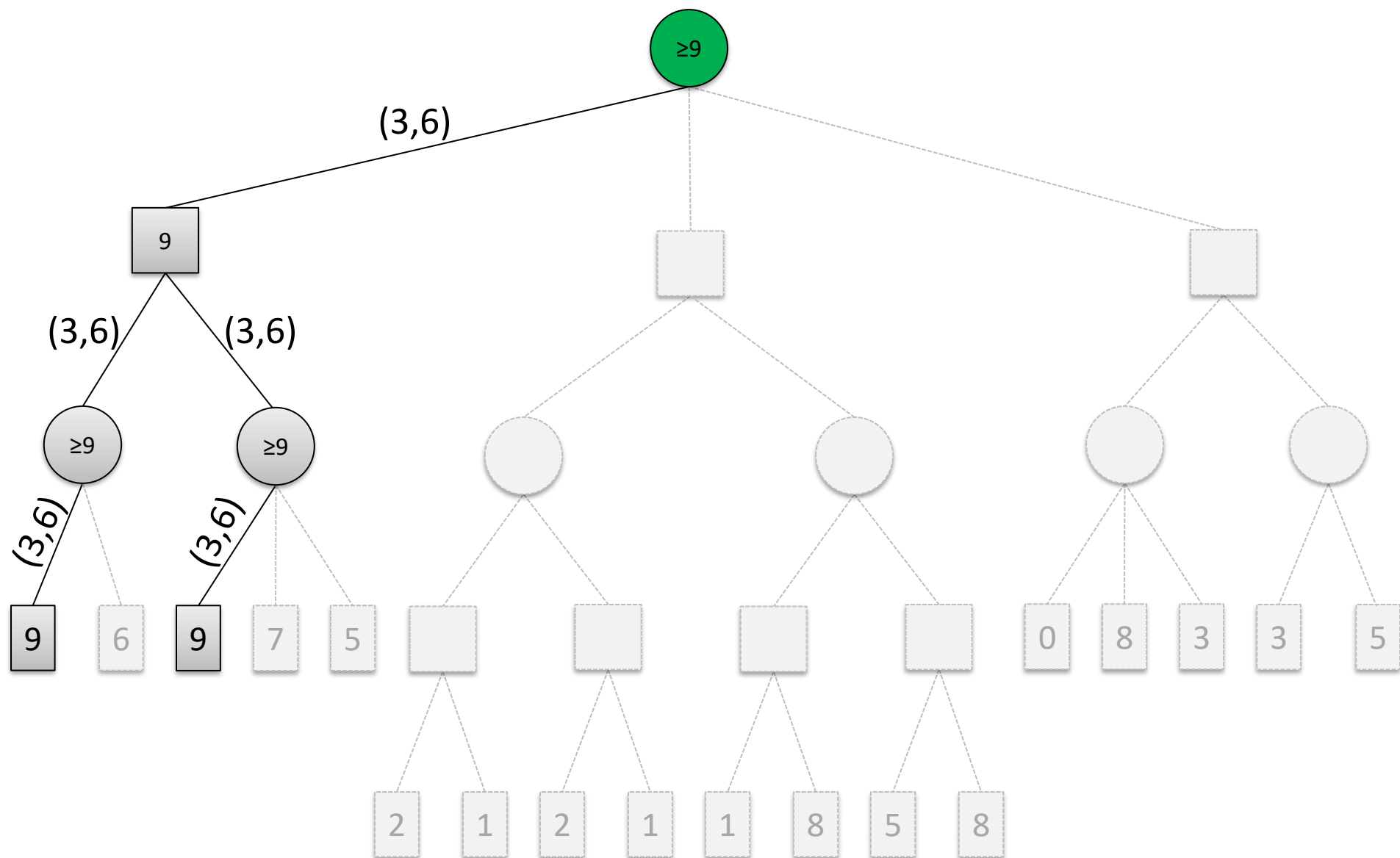


Legenda:  ruch gracza Max  ruch gracza Min



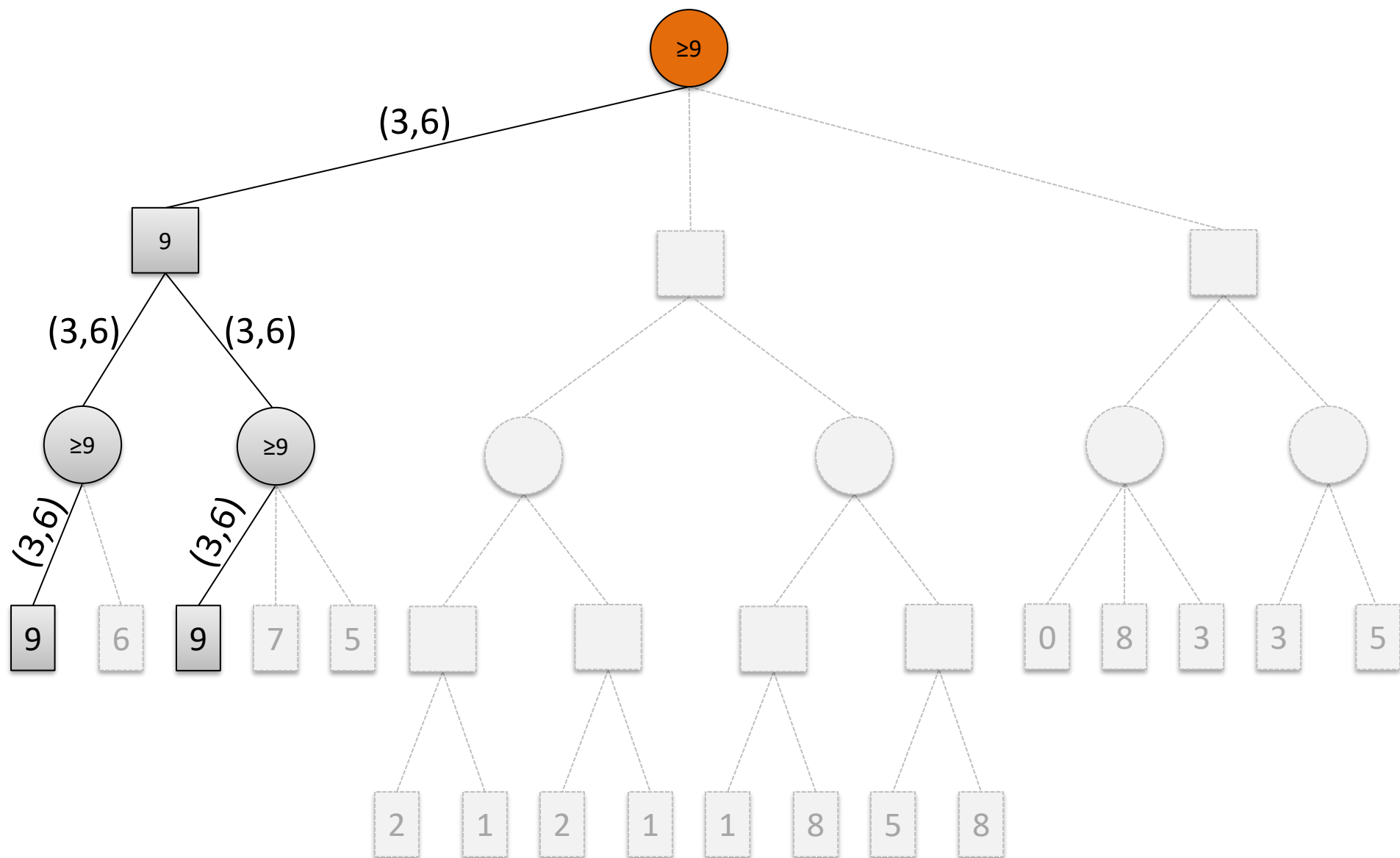
Legenda:  ruch gracza Max  ruch gracza Min





Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max

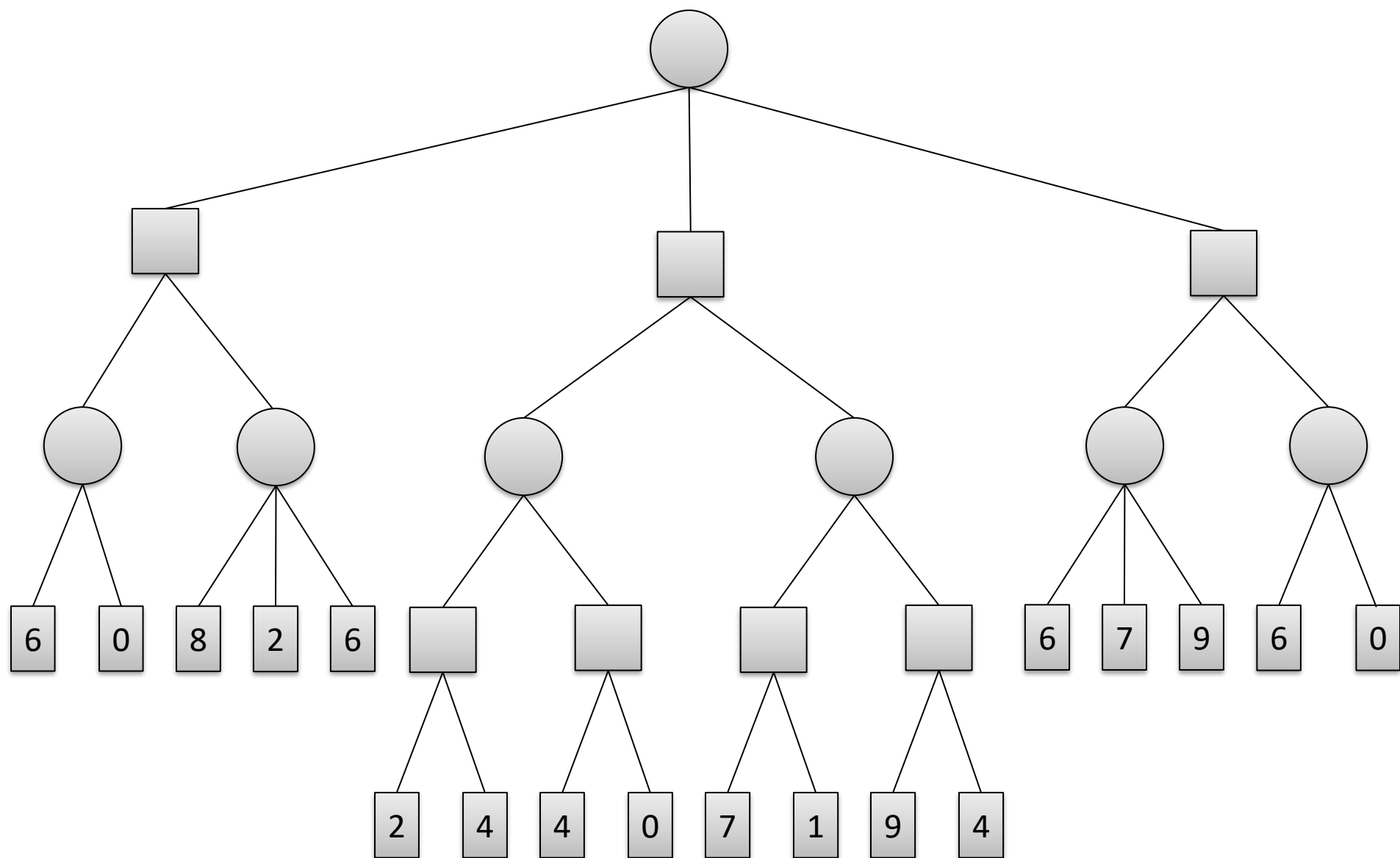
 ruch gracza Min

Niech $(\mathbb{S}, s, N, ev)$ – gra, $s \in \mathbb{S}$ – stan gry, $0 \leq \alpha < \beta \leq 1$.
Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz
 $m = \text{MinMax}(s)$.

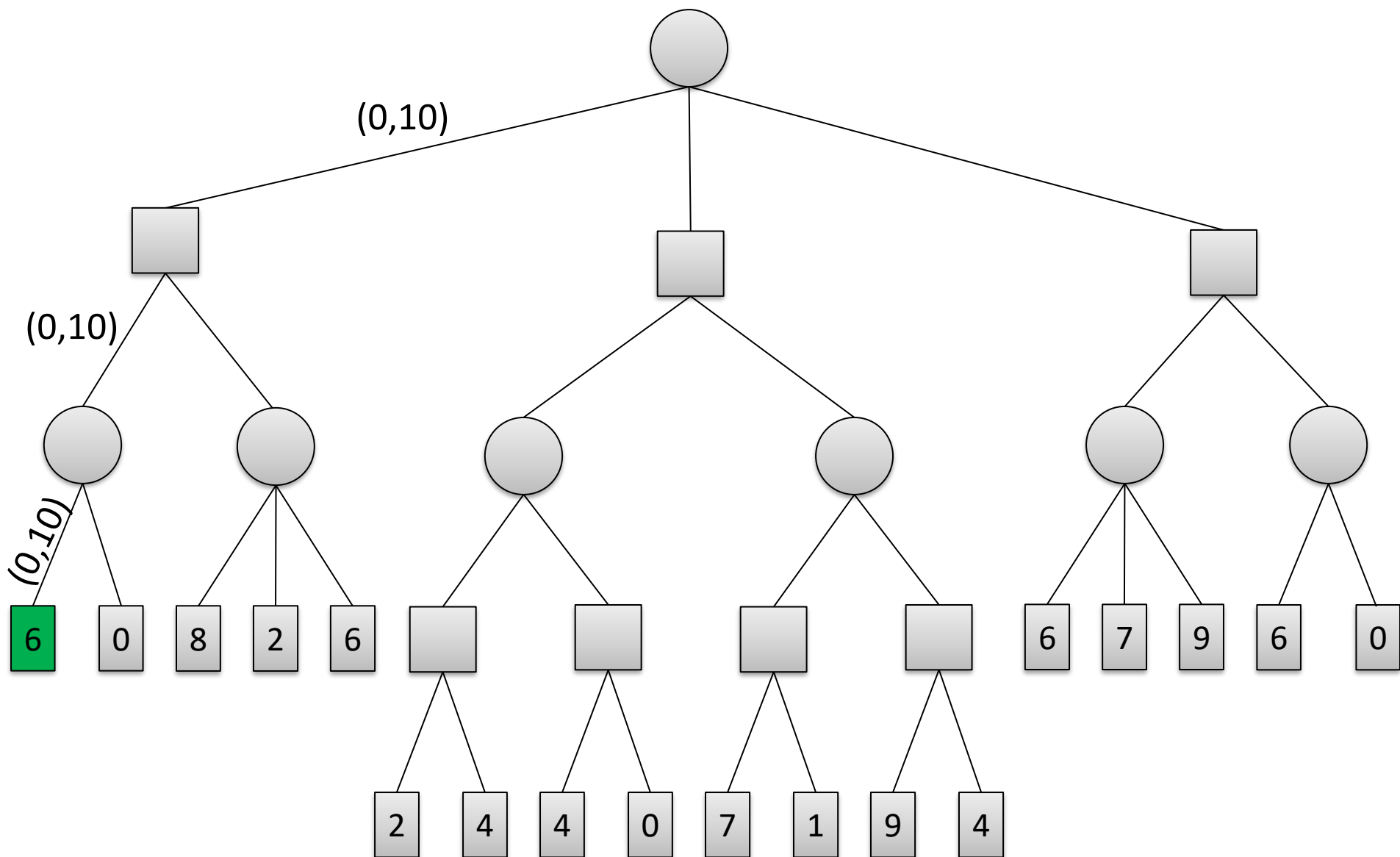
Algorytm Alfa – Beta zwraca wartość powyżej β .
 $\beta \leq ab \Rightarrow ab \leq m$



Przykład 2. Rozważmy działanie wywołanego algorytmu α - β w oknie (0,10) poniższego grafu gry:



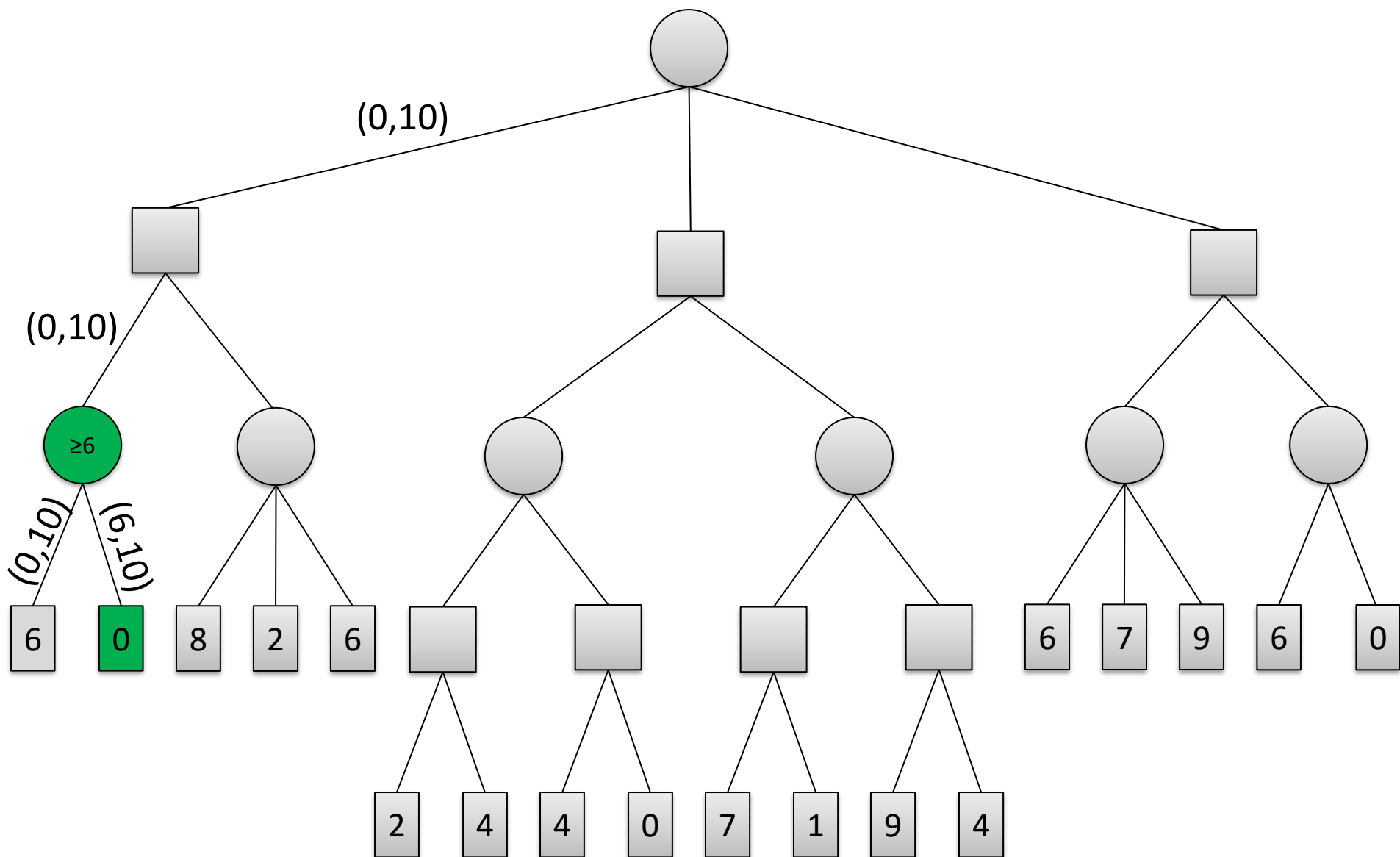




Legenda:  ruch gracza Max

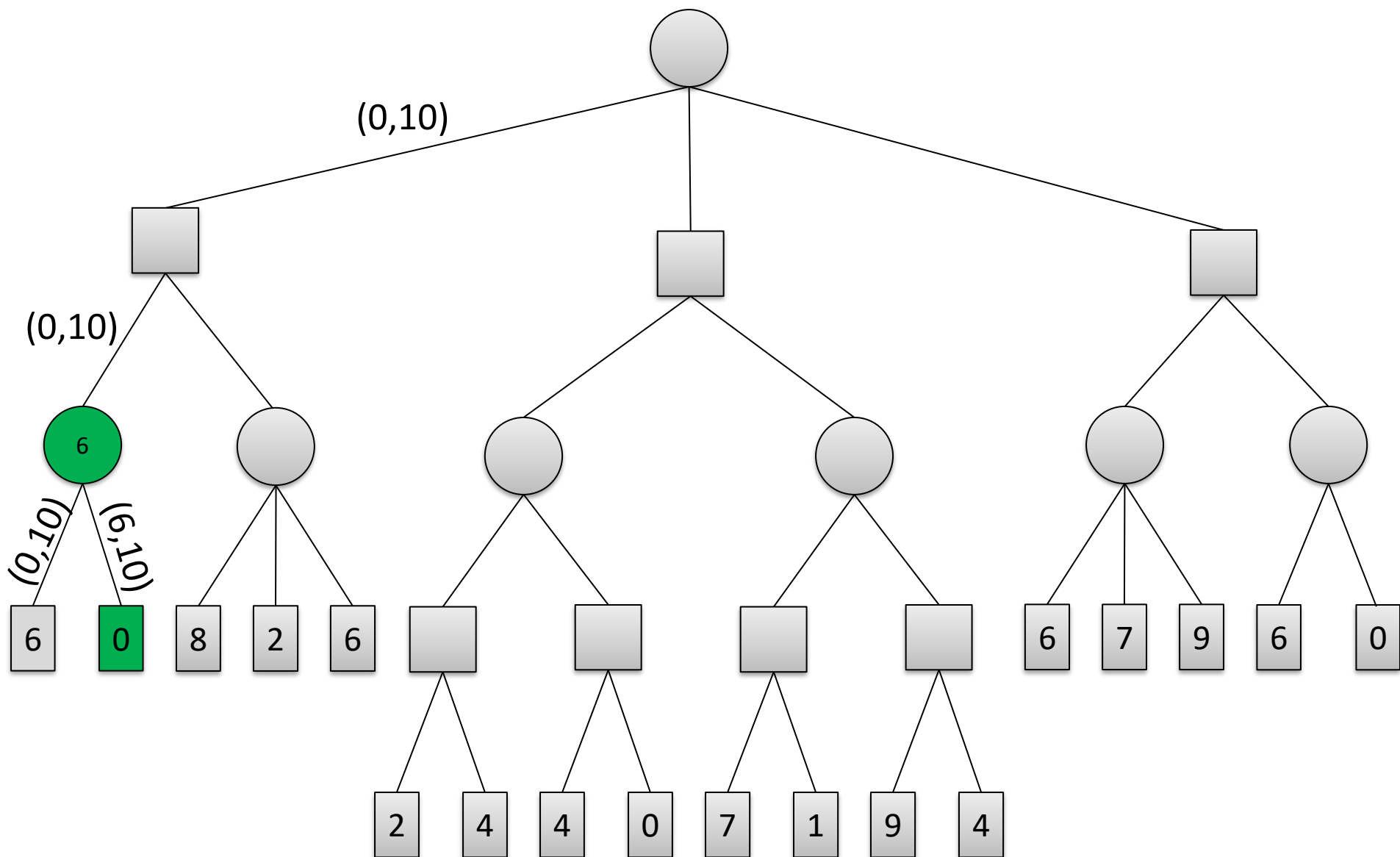
 ruch gracza Min





Legenda:  ruch gracza Max

 ruch gracza Min

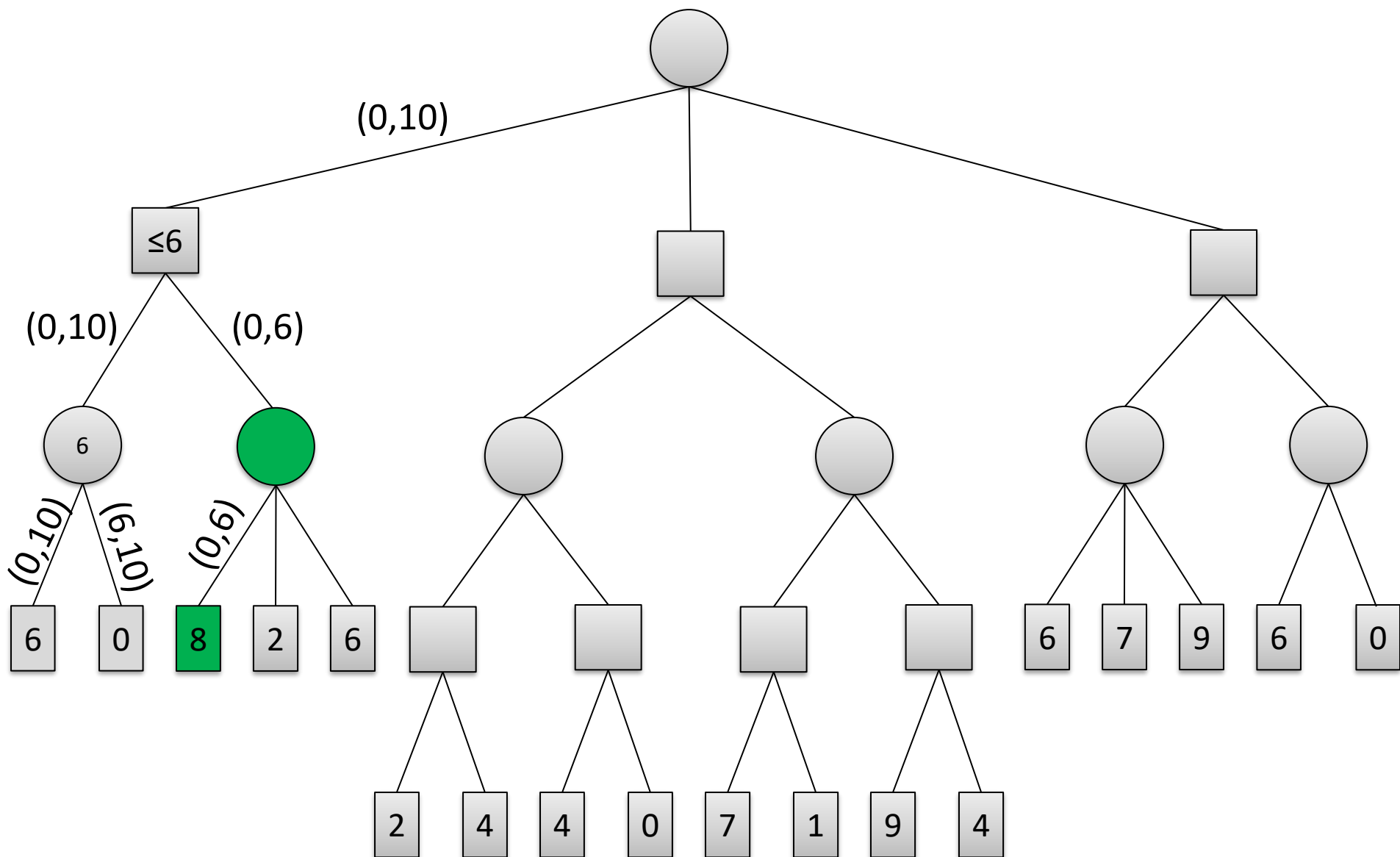


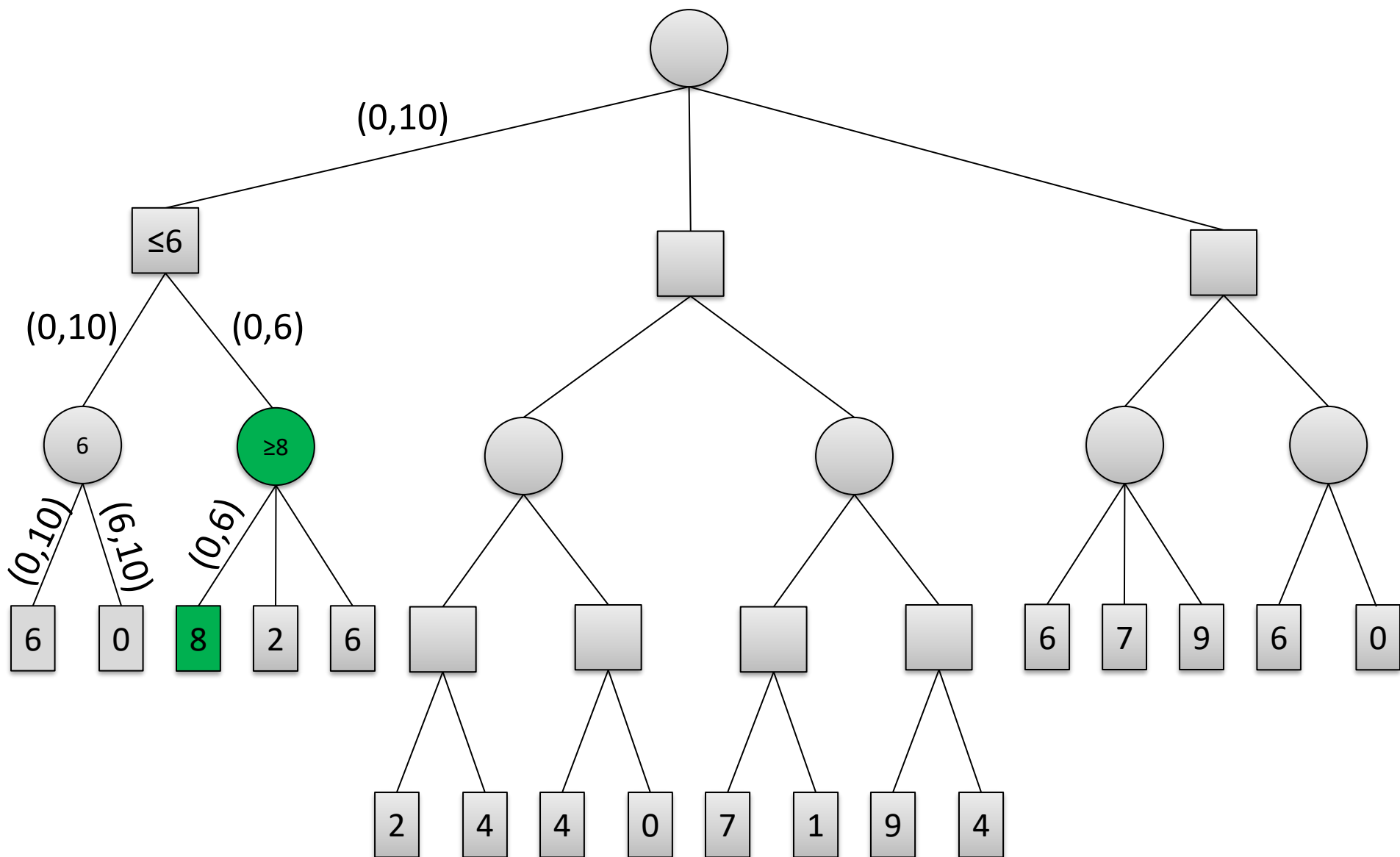
Legenda:  ruch gracza Max

 ruch gracza Min



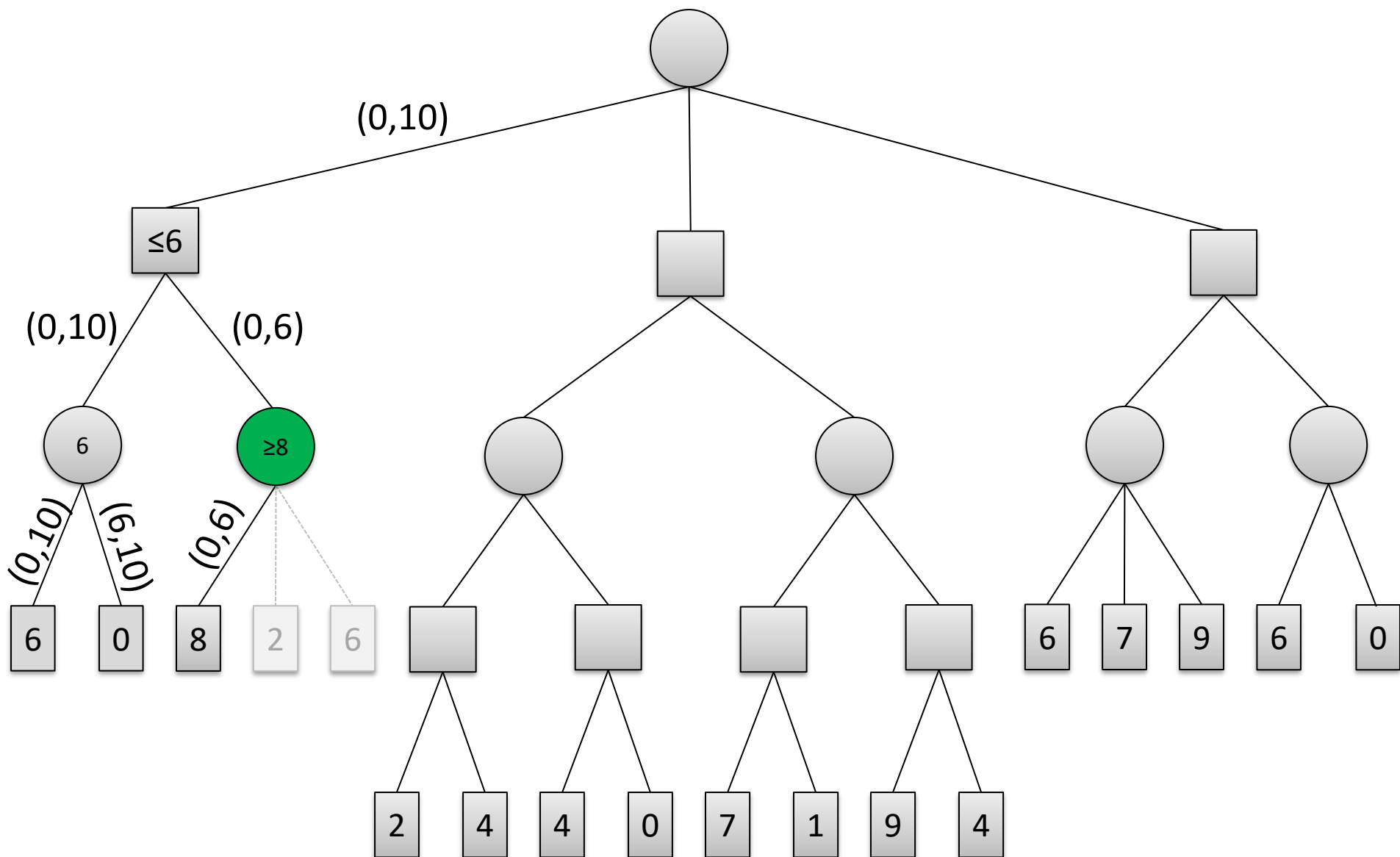






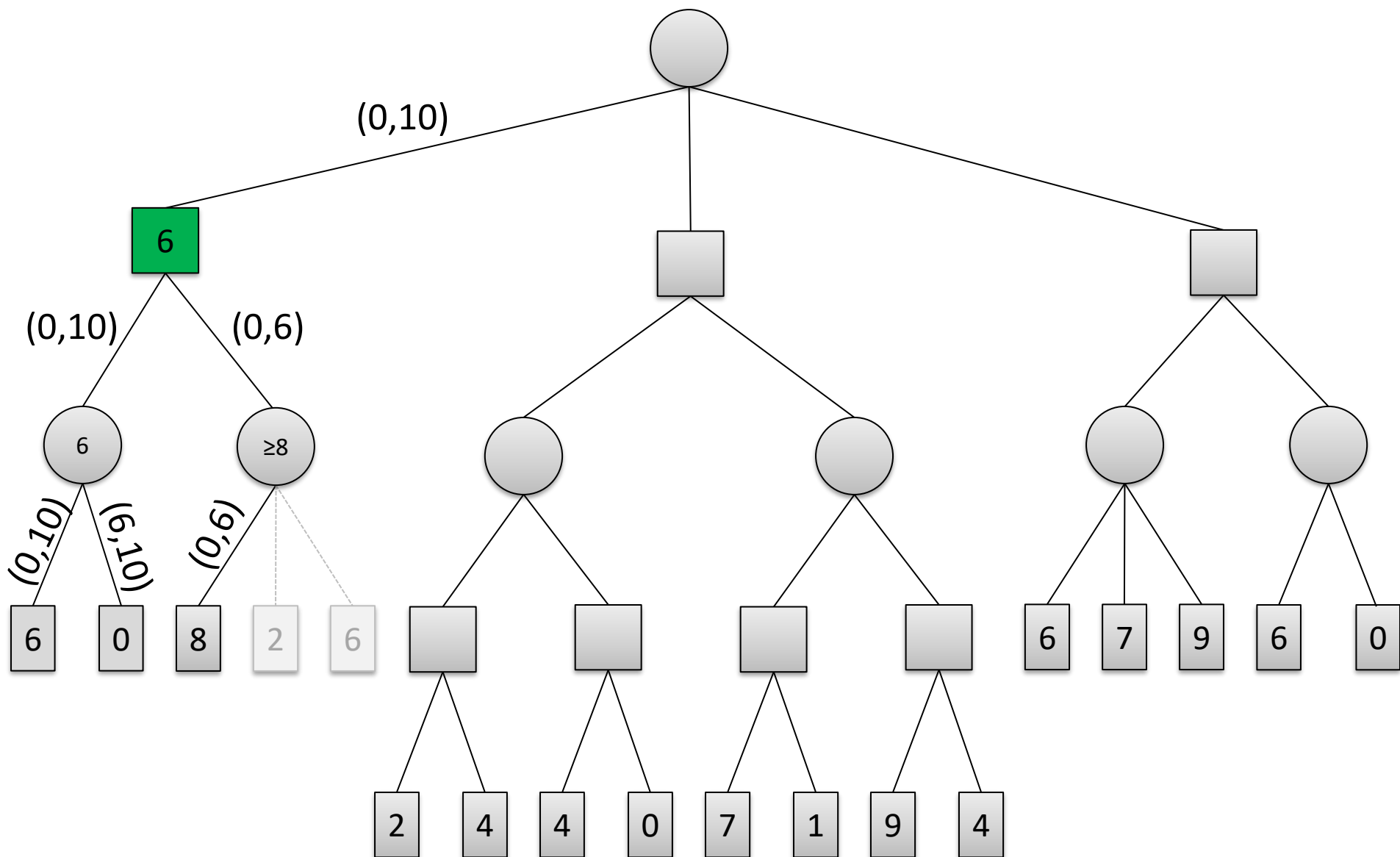
Legenda:  ruch gracza Max

 ruch gracza Min



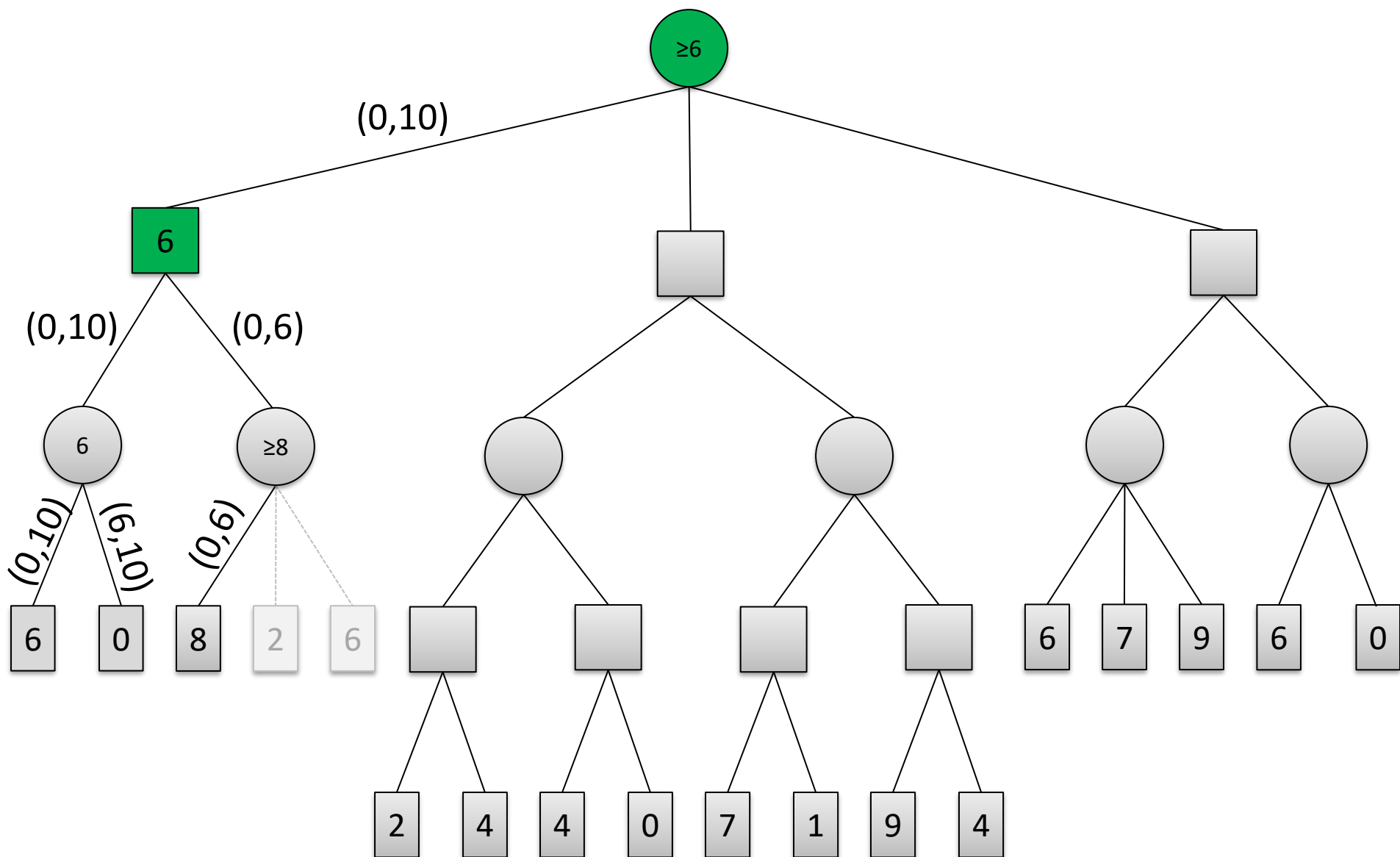
Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max

 ruch gracza Min

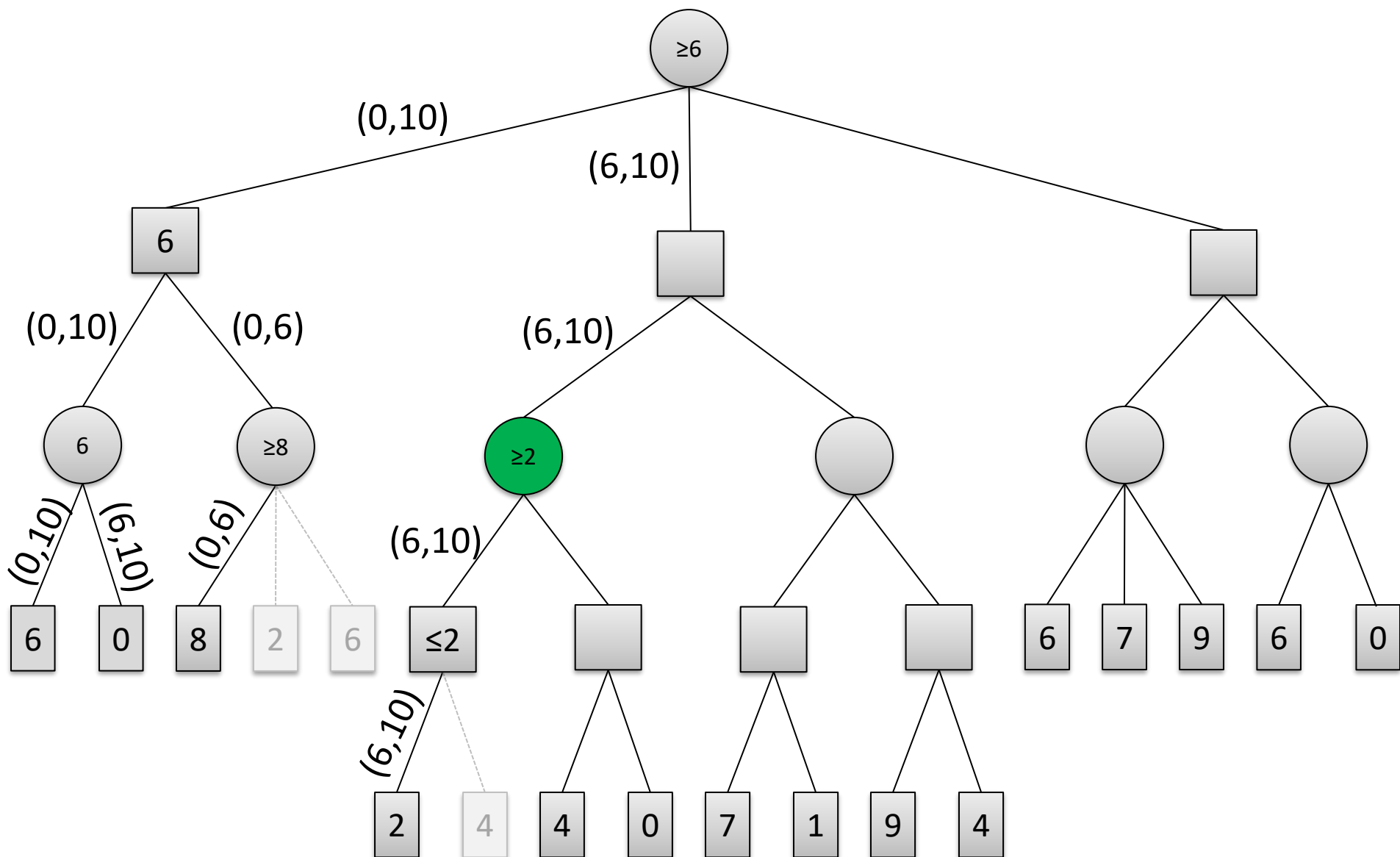


Legenda:  ruch gracza Max

 ruch gracza Min

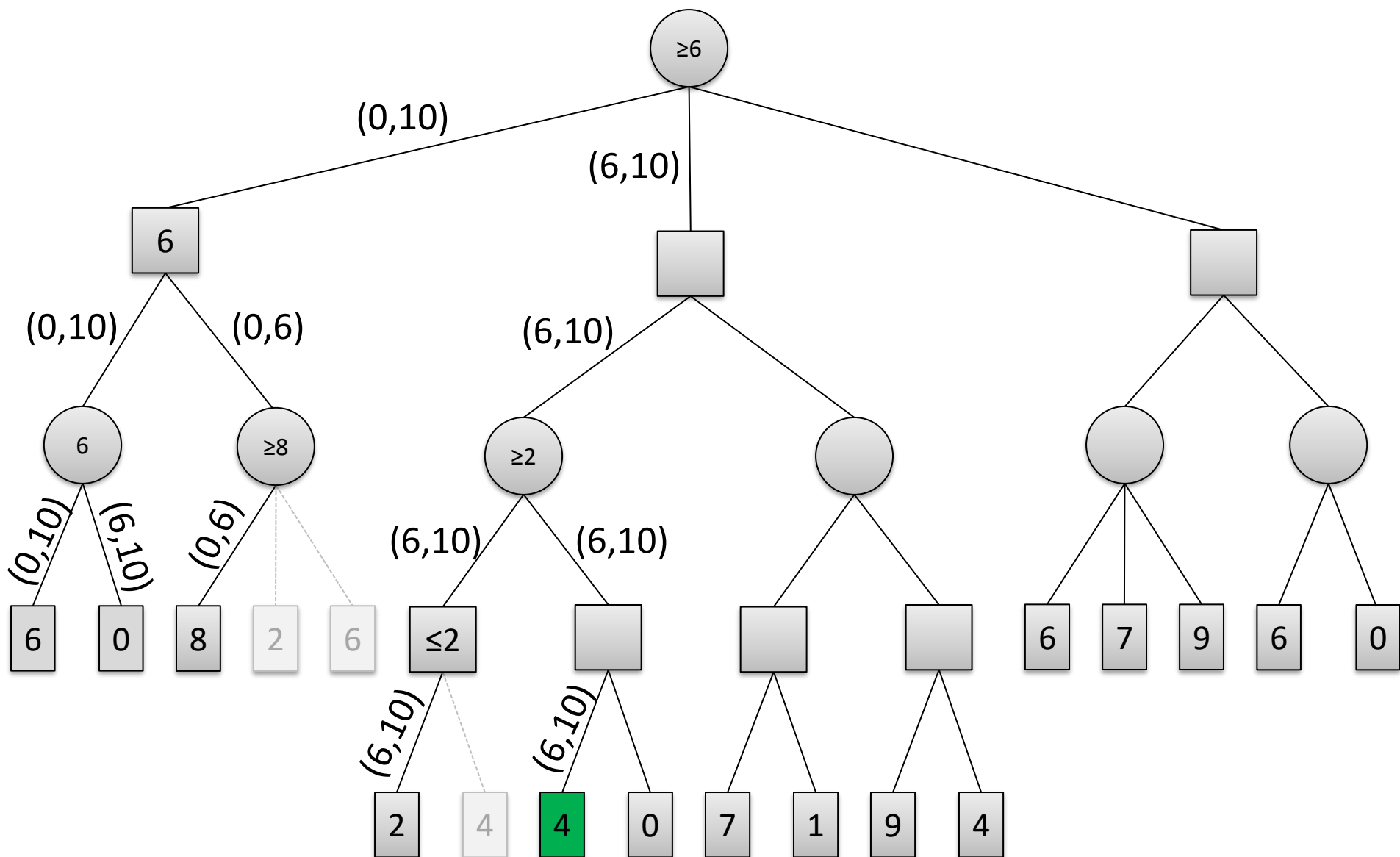






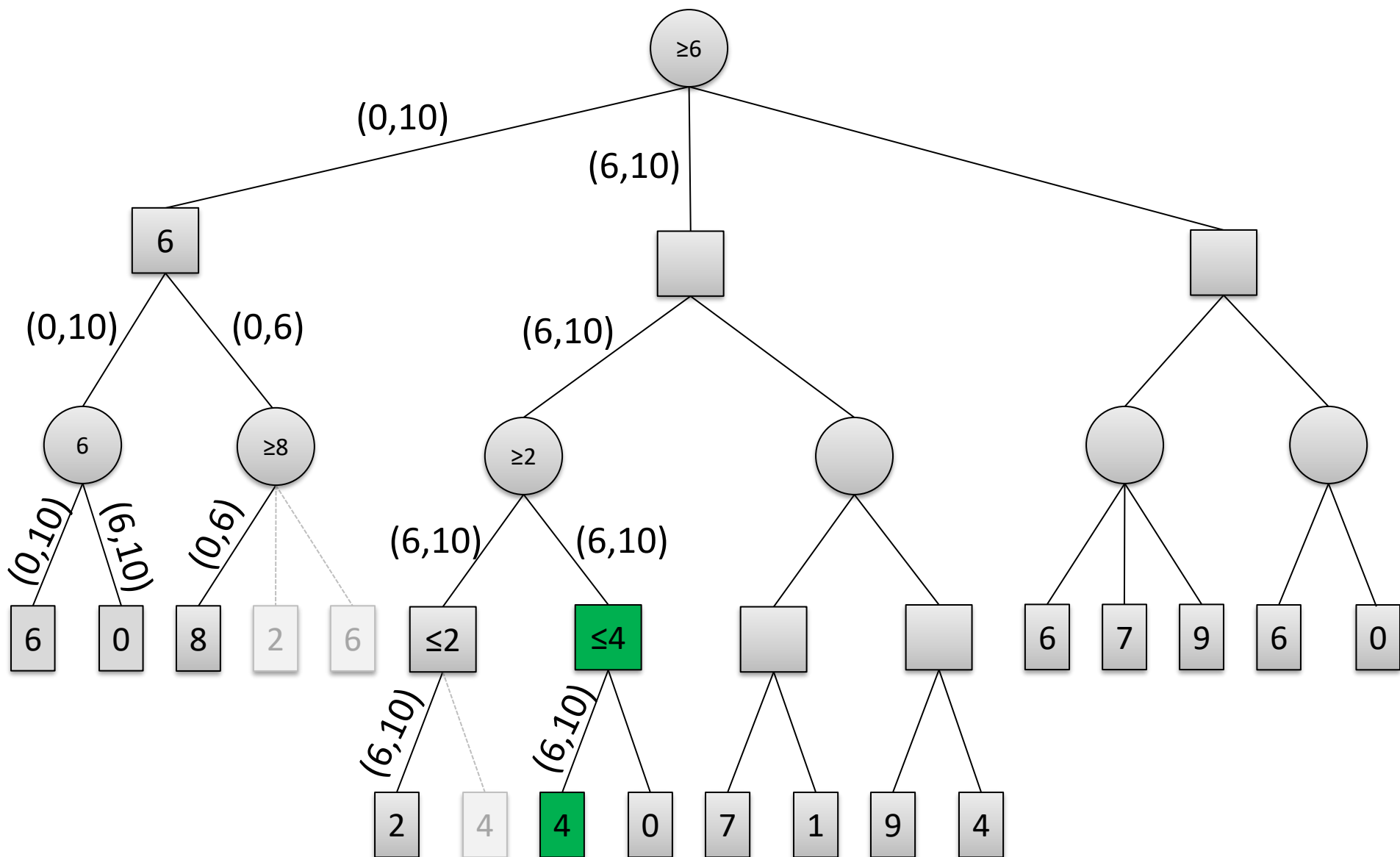
Legenda:  ruch gracza Max

 ruch gracza Min



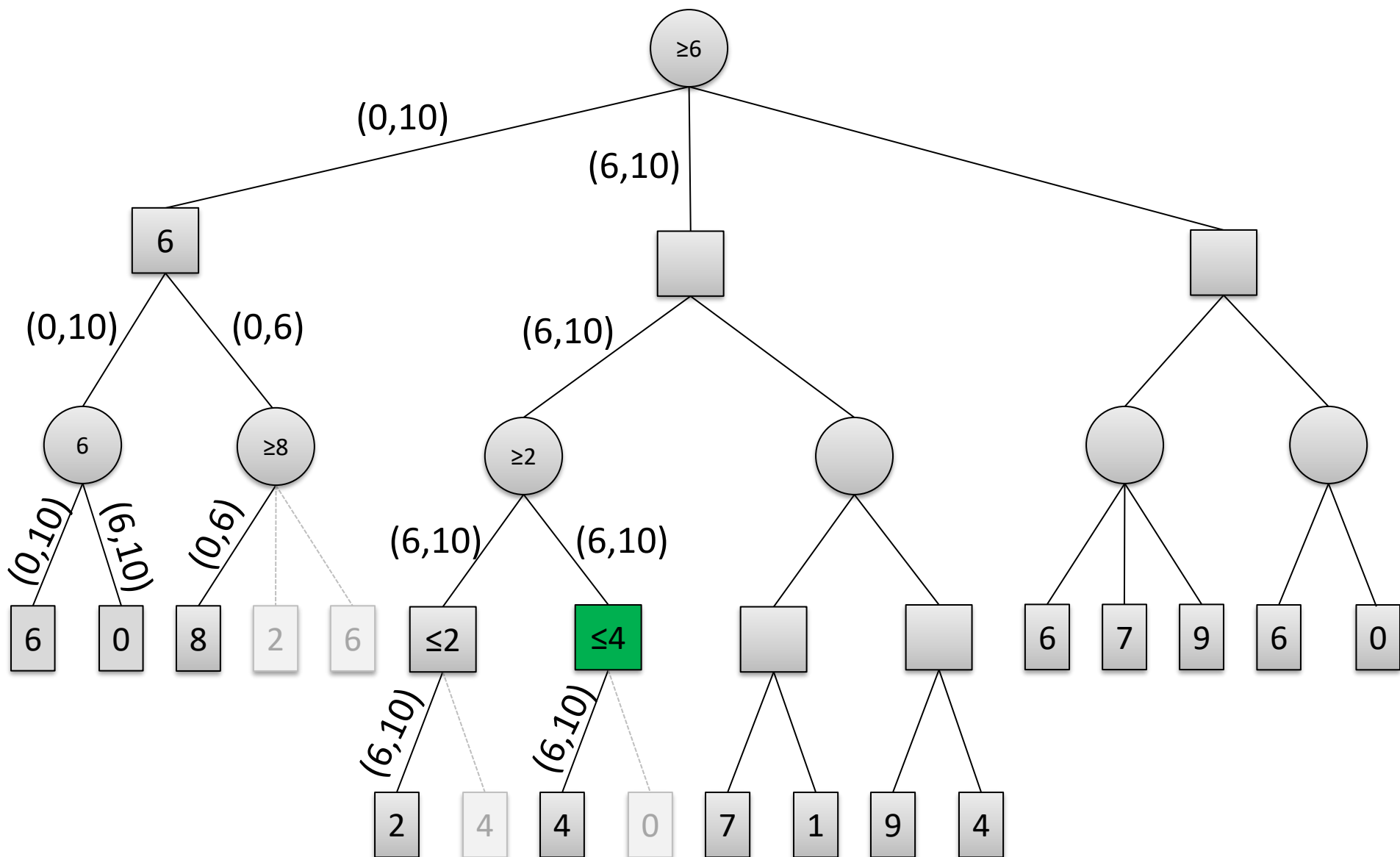
Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max

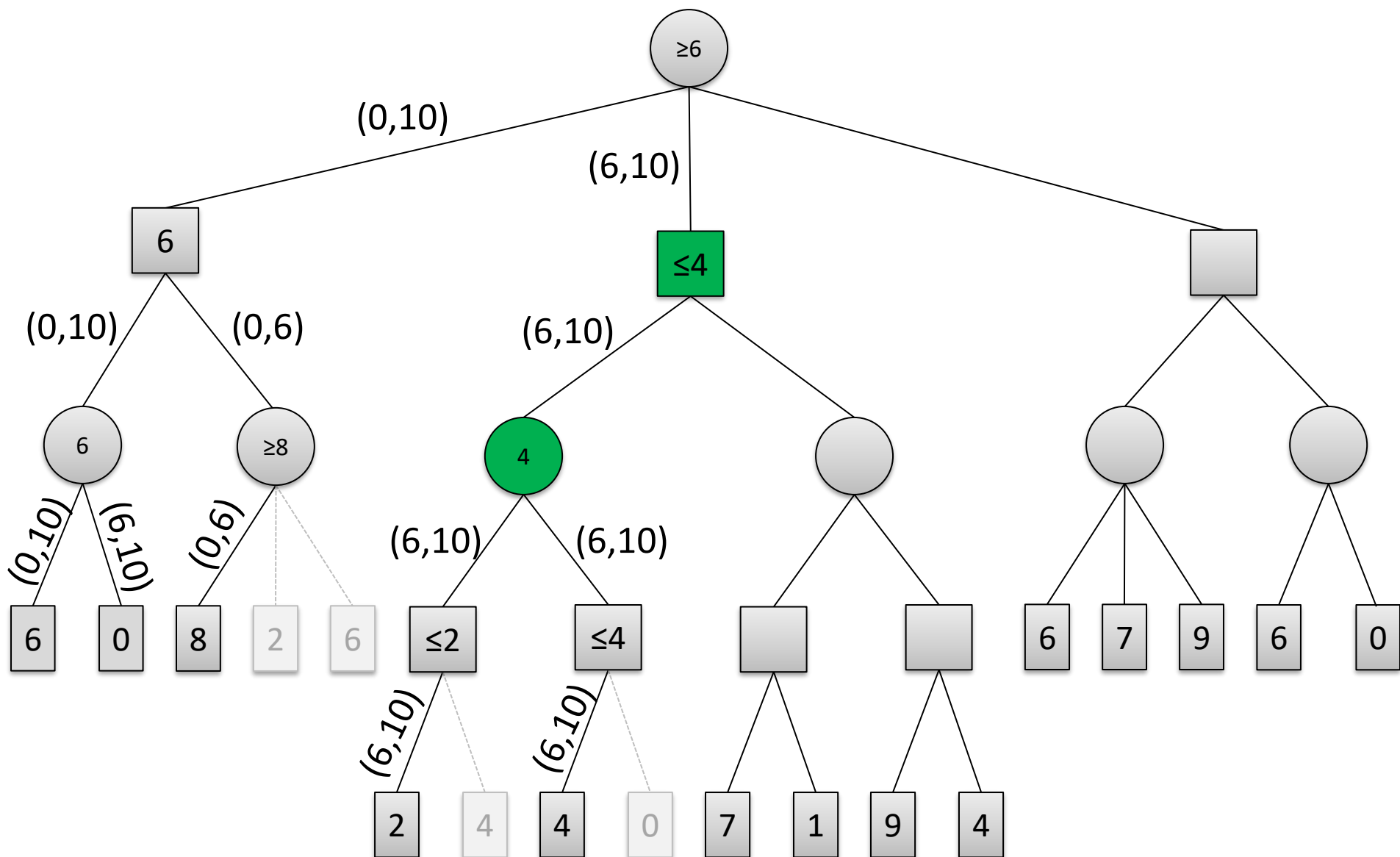
 ruch gracza Min



Legenda:  ruch gracza Max

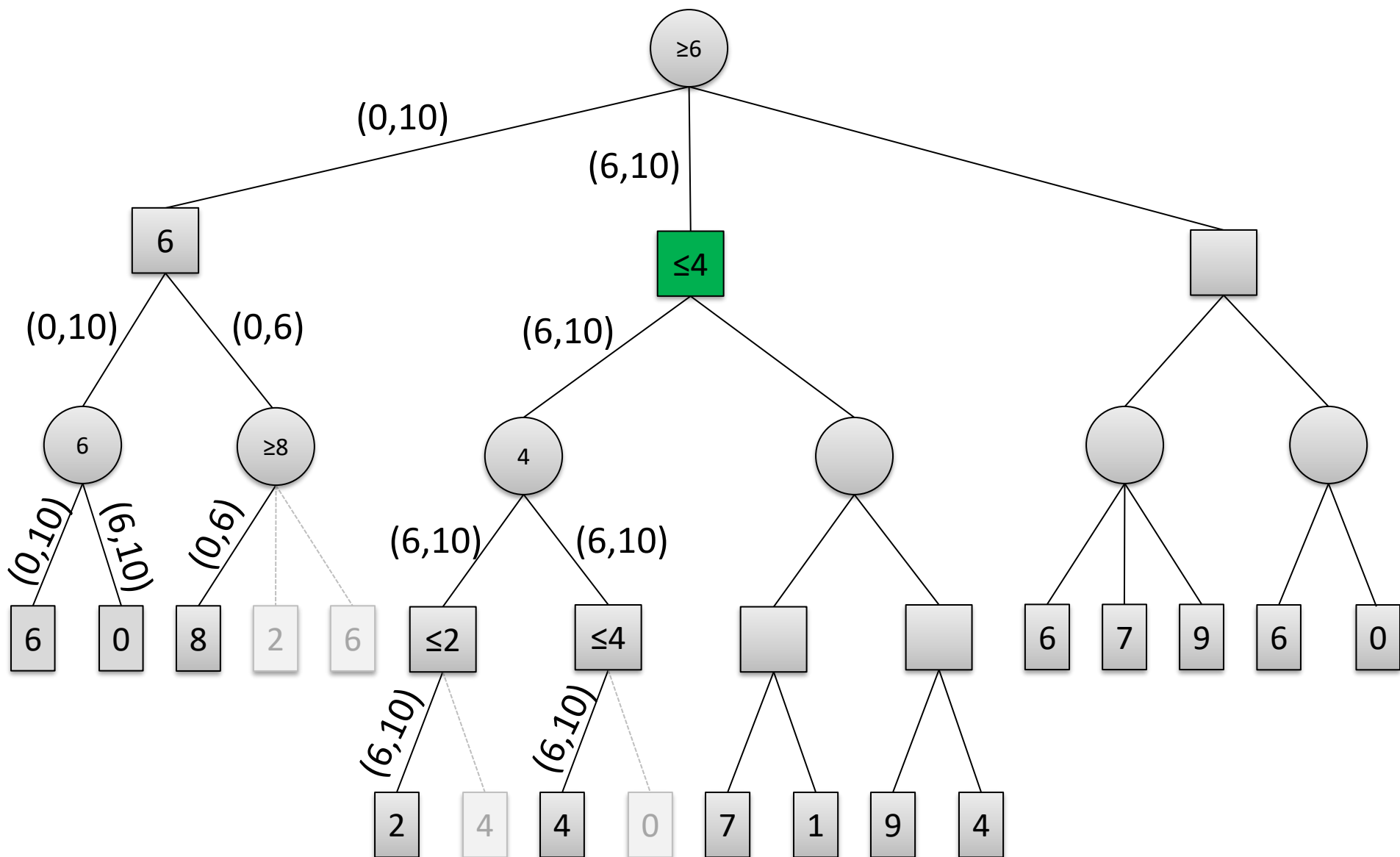
 ruch gracza Min





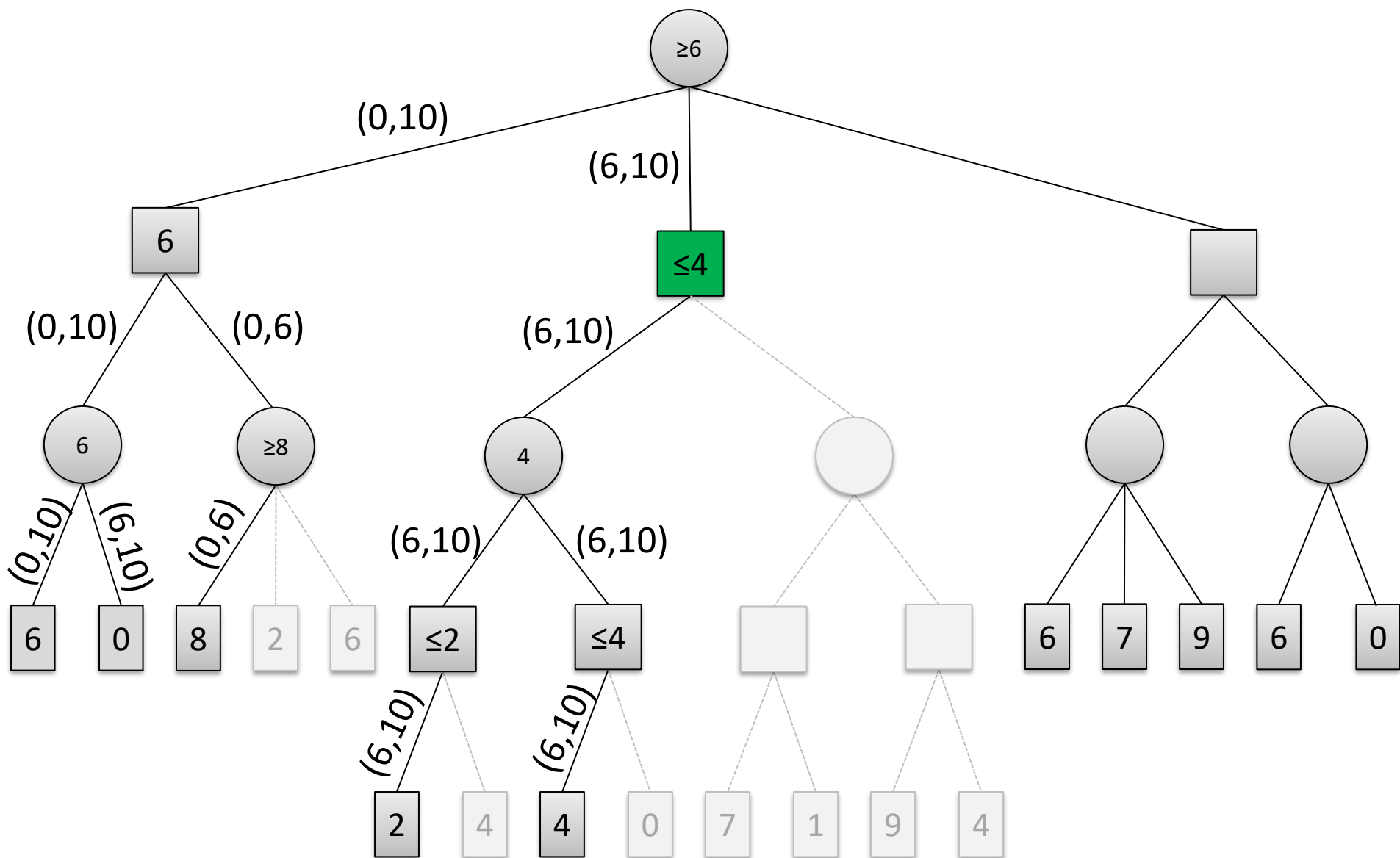
Legenda:  ruch gracza Max

 ruch gracza Min

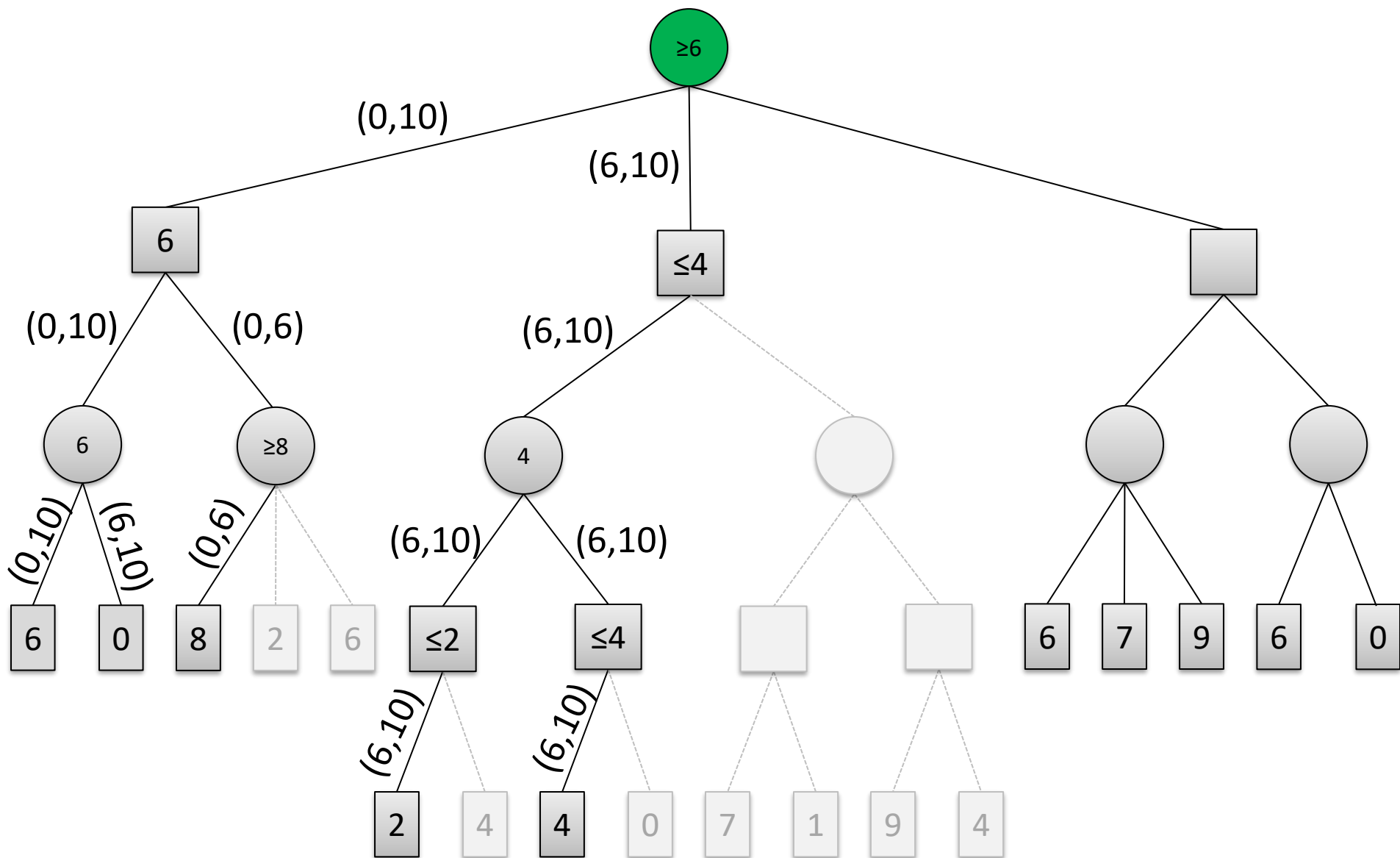


Legenda:  ruch gracza Max

 ruch gracza Min

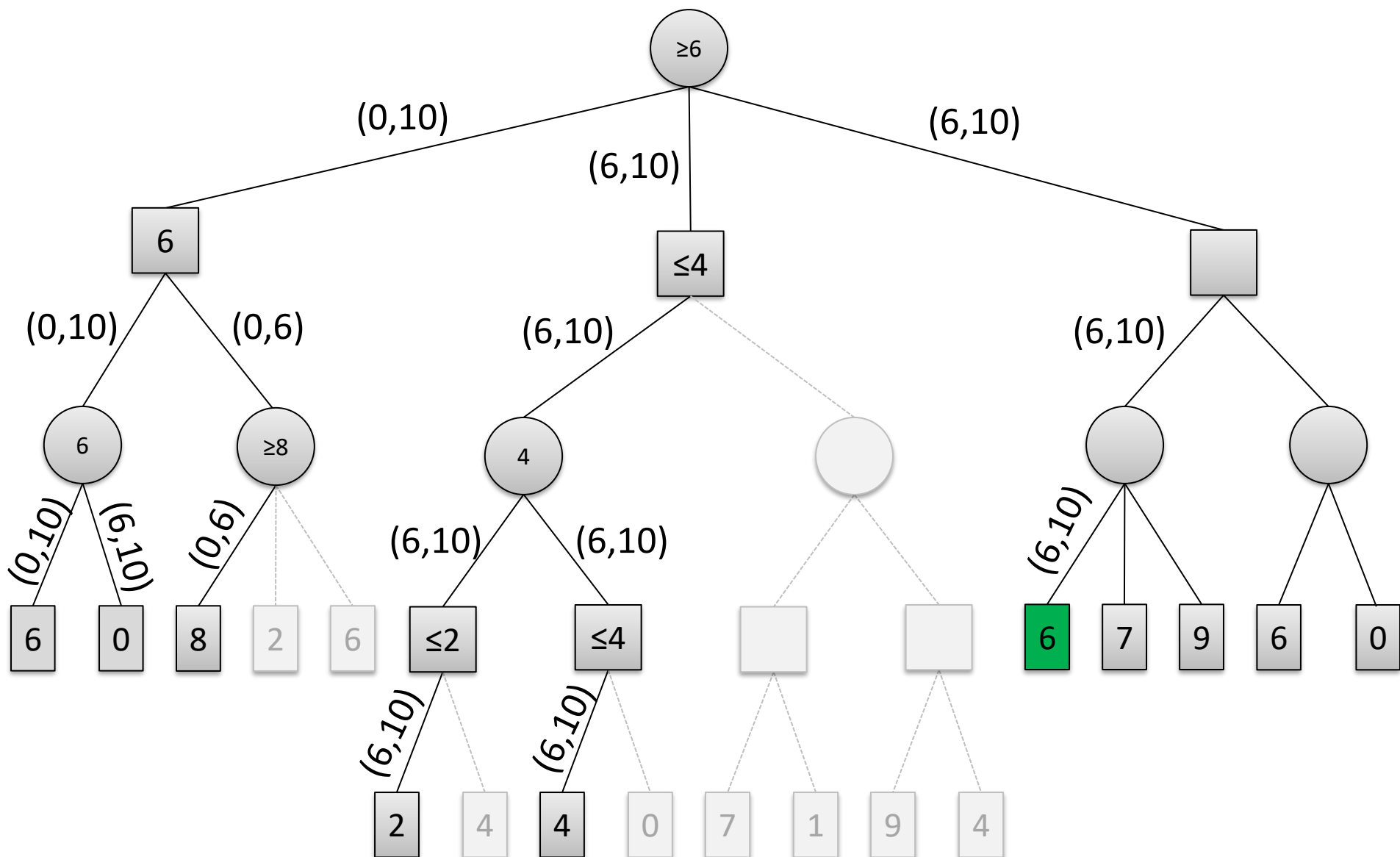


Legenda:  ruch gracza Max  ruch gracza Min

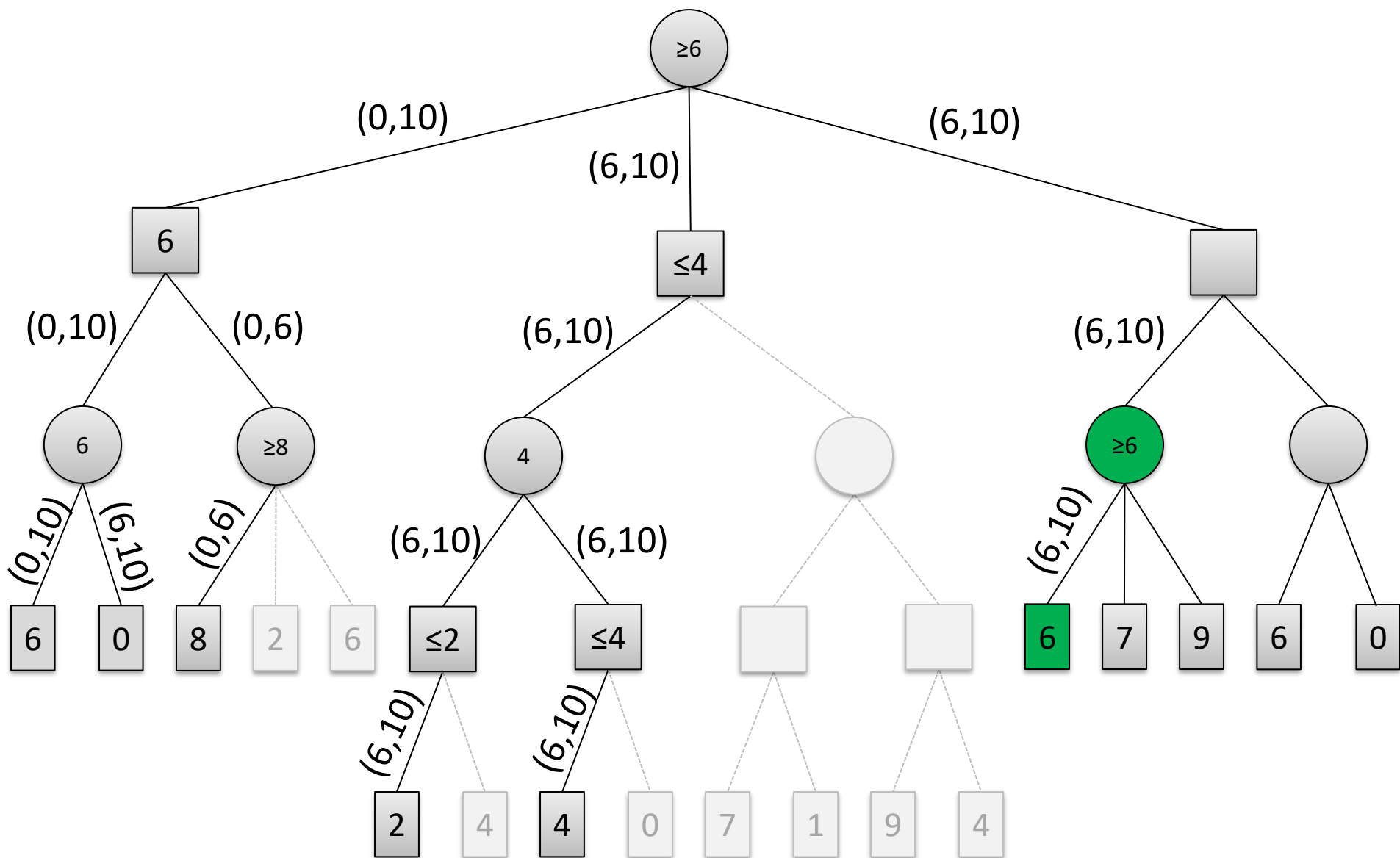


Legenda:  ruch gracza Max

 ruch gracza Min



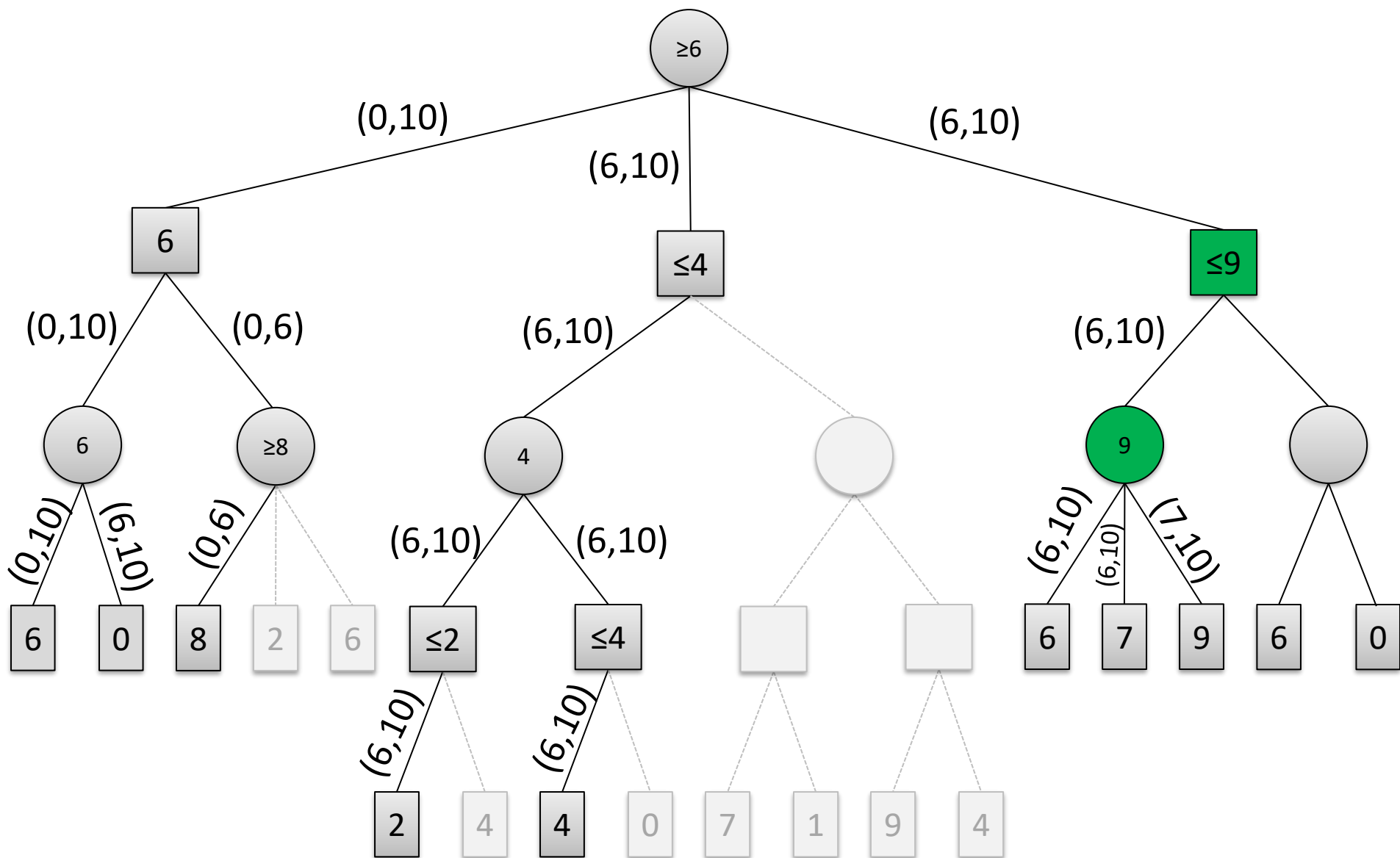
Legenda:  ruch gracza Max  ruch gracza Min



Legenda:  ruch gracza Max  ruch gracza Min

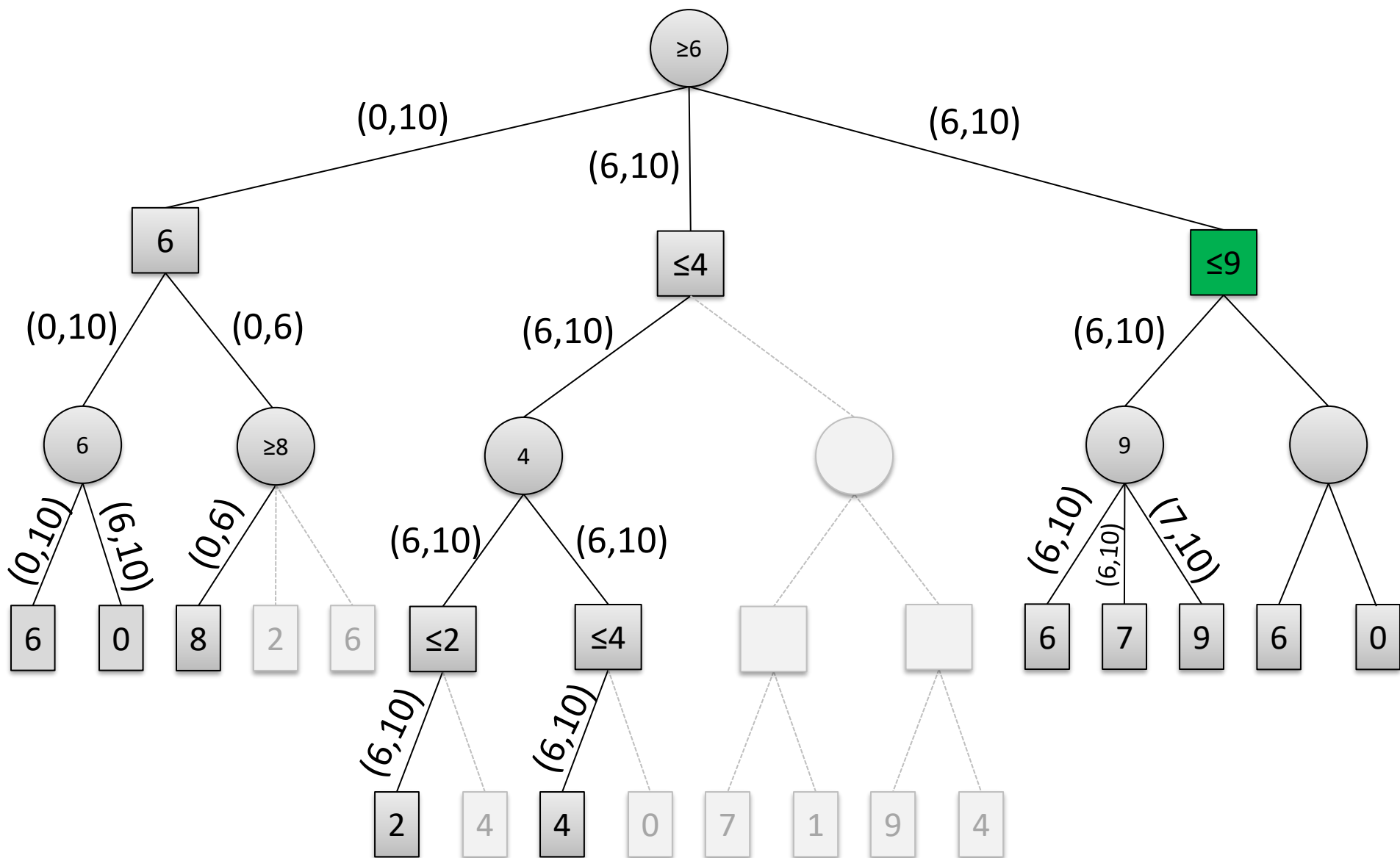






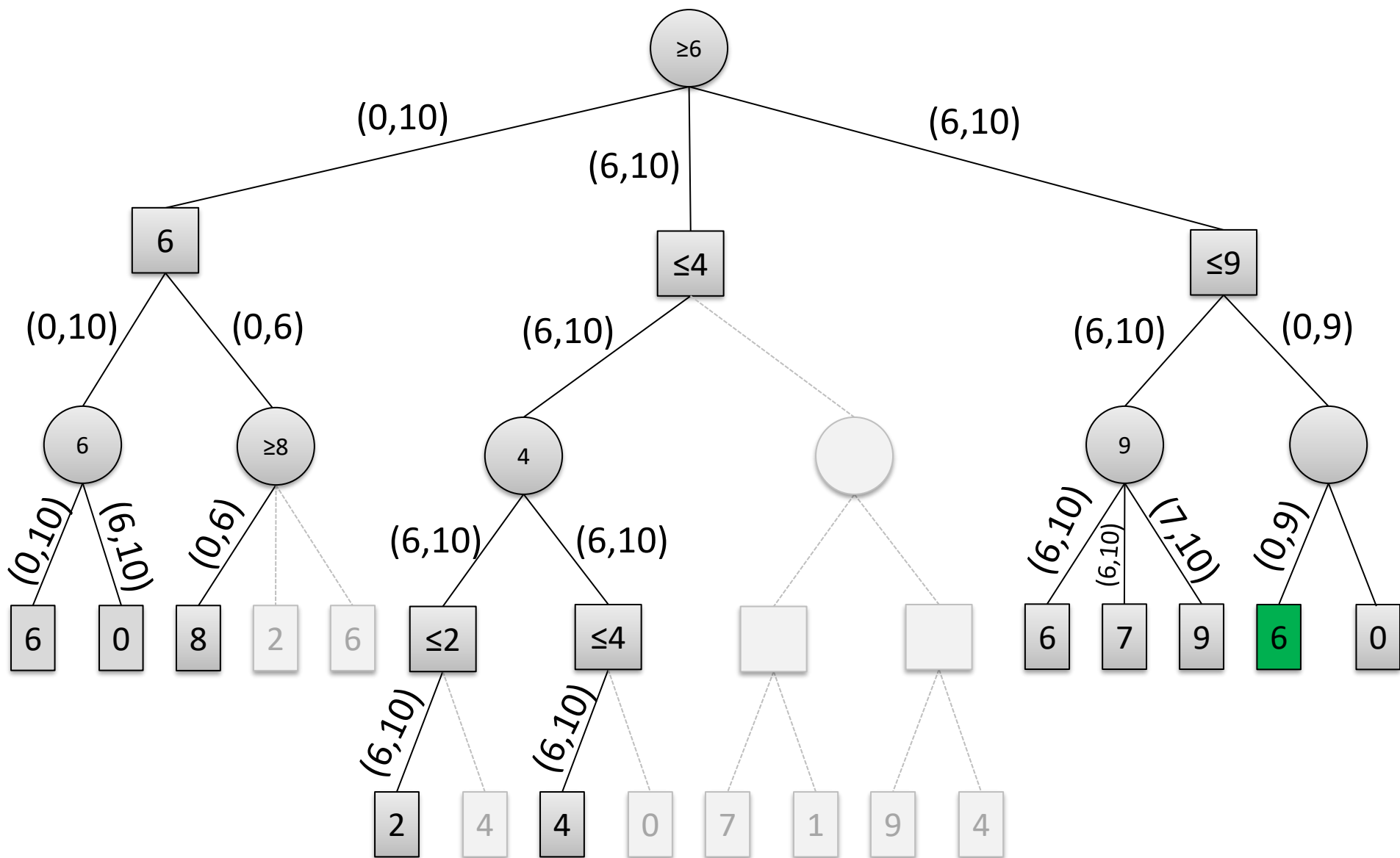
Legenda:  ruch gracza Max

 ruch gracza Min

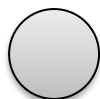


Legenda:  ruch gracza Max

 ruch gracza Min



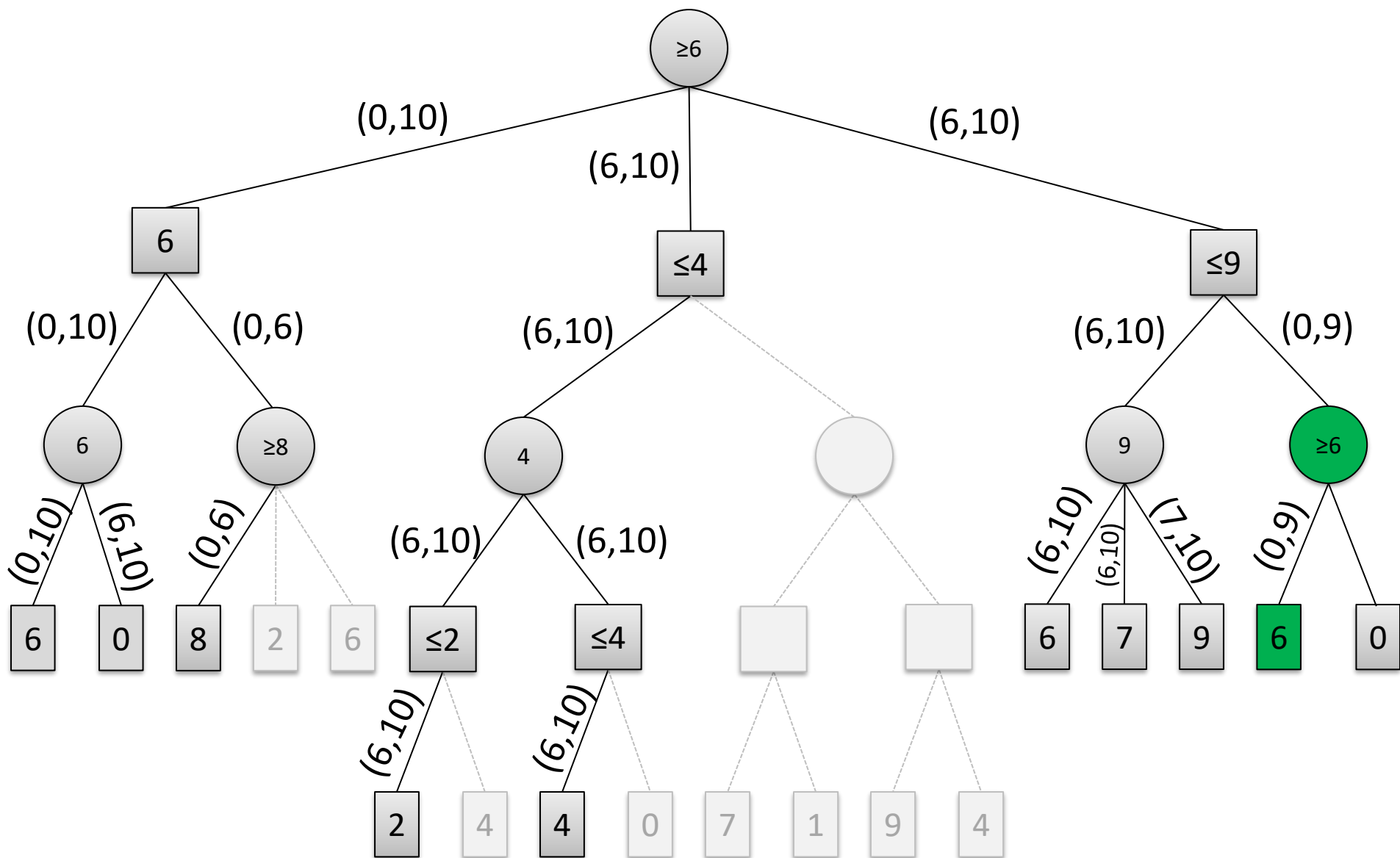
Legenda:



ruch gracza Max

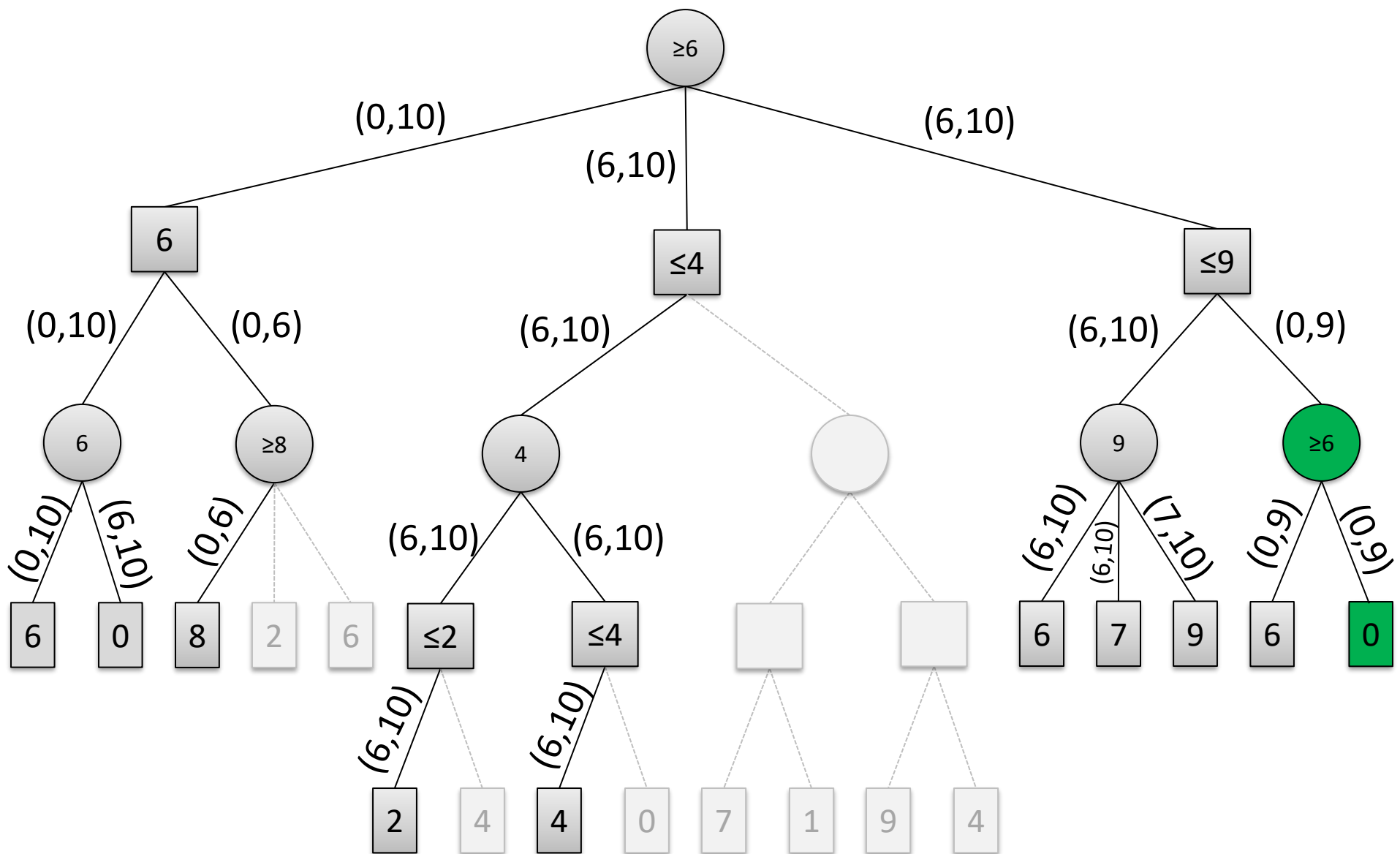


ruch gracza Min



Legenda:  ruch gracza Max

 ruch gracza Min



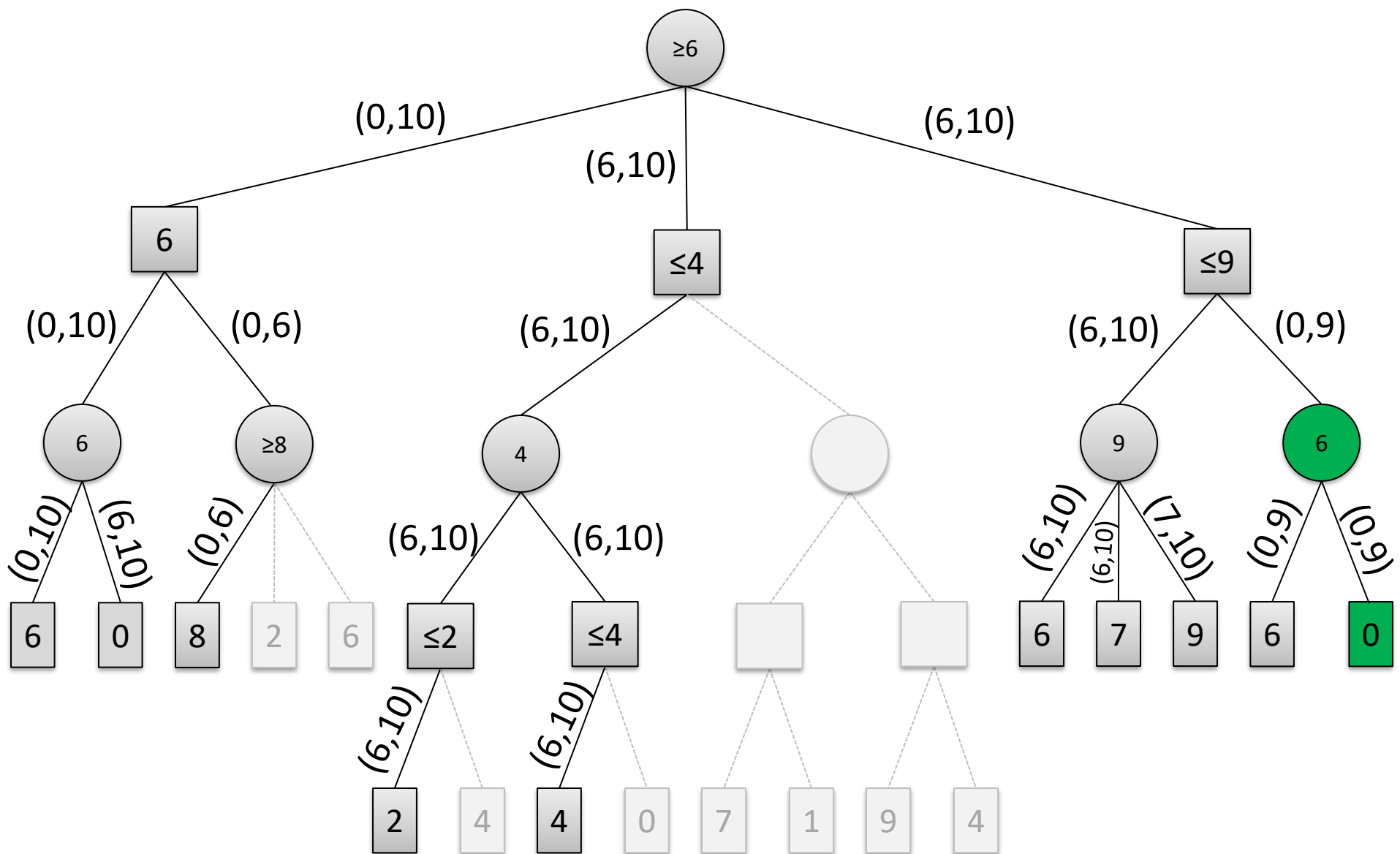
Legenda:



ruch gracza Max

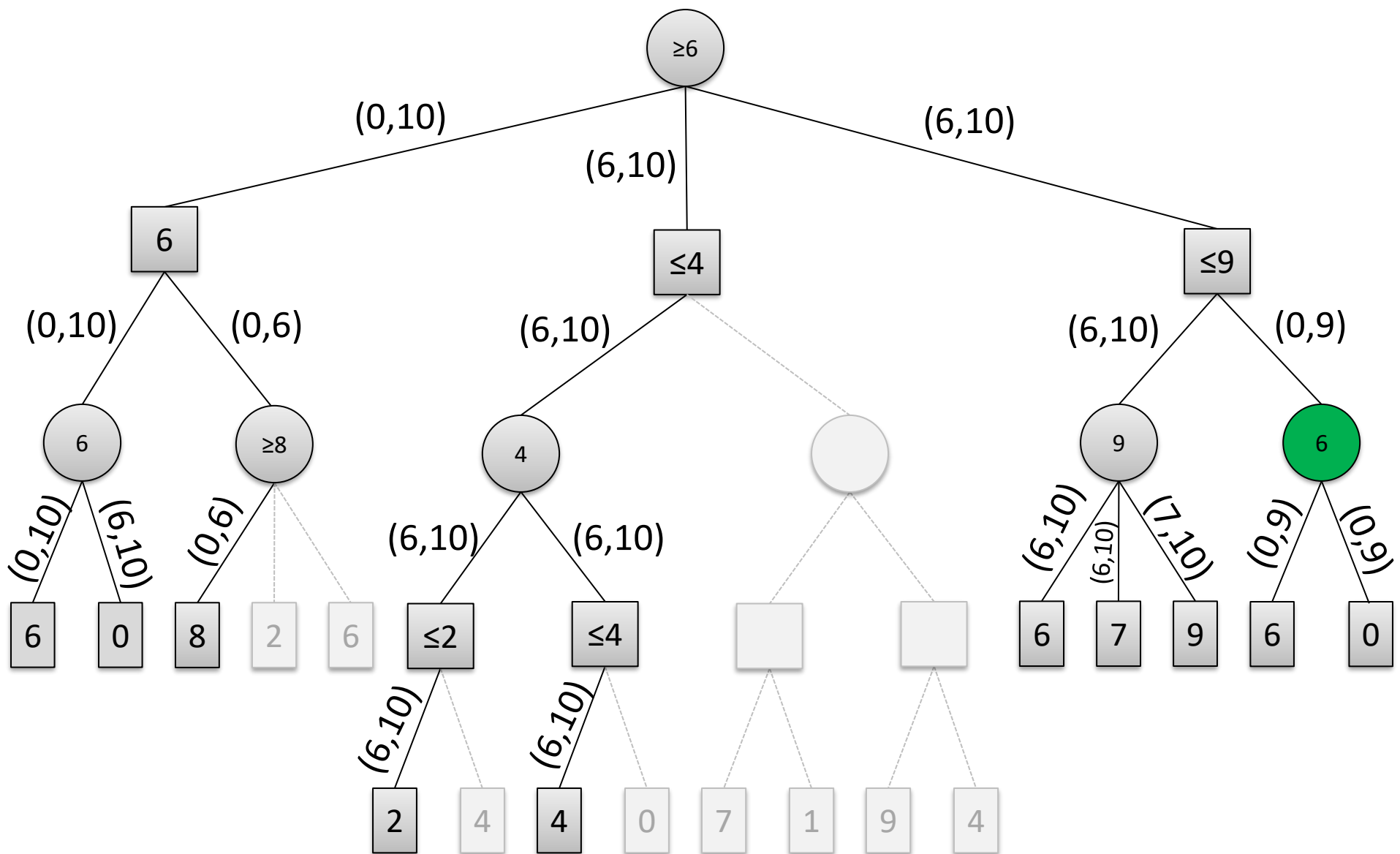


ruch gracza Min



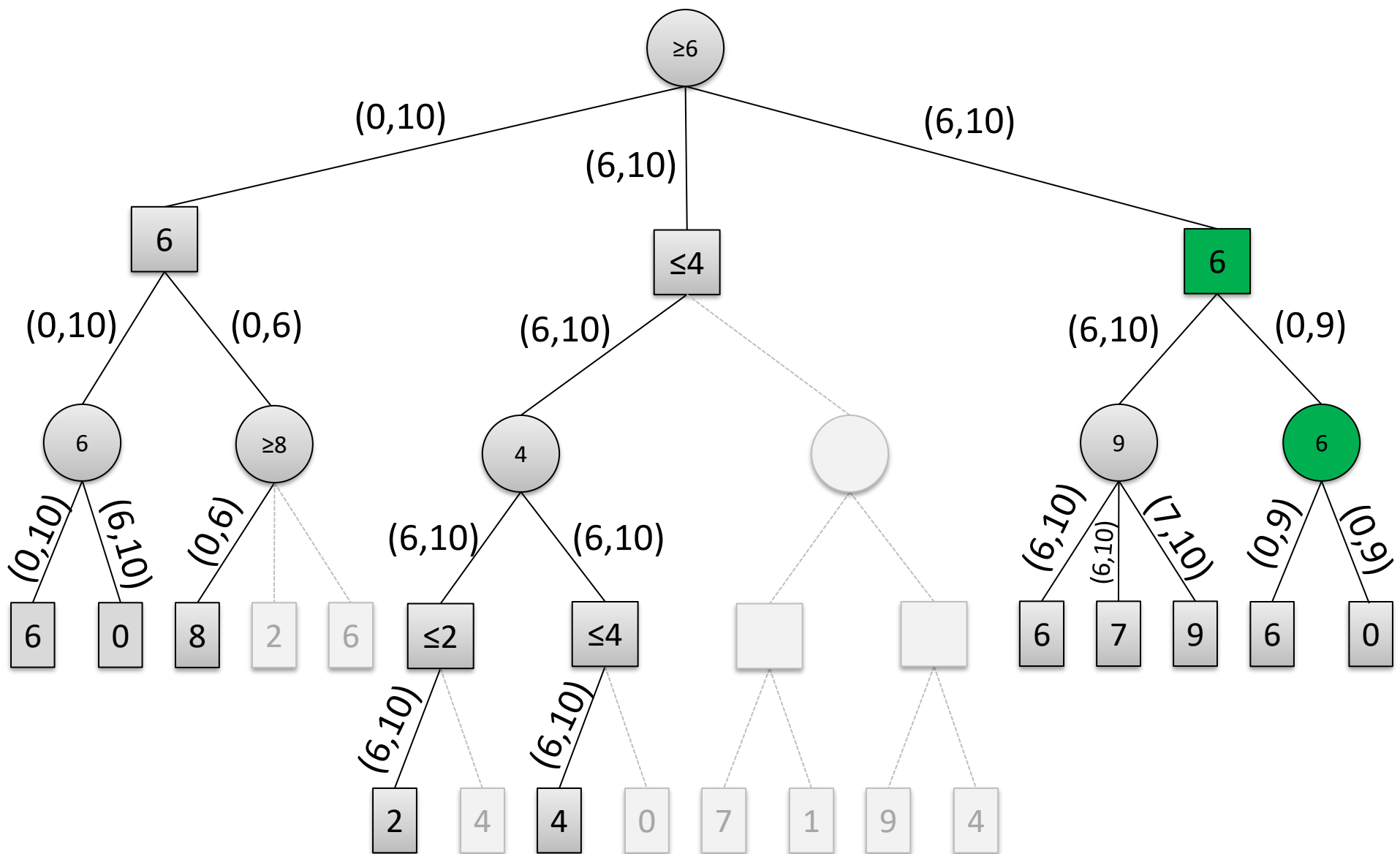
Legenda:  ruch gracza Max

 ruch gracza Min



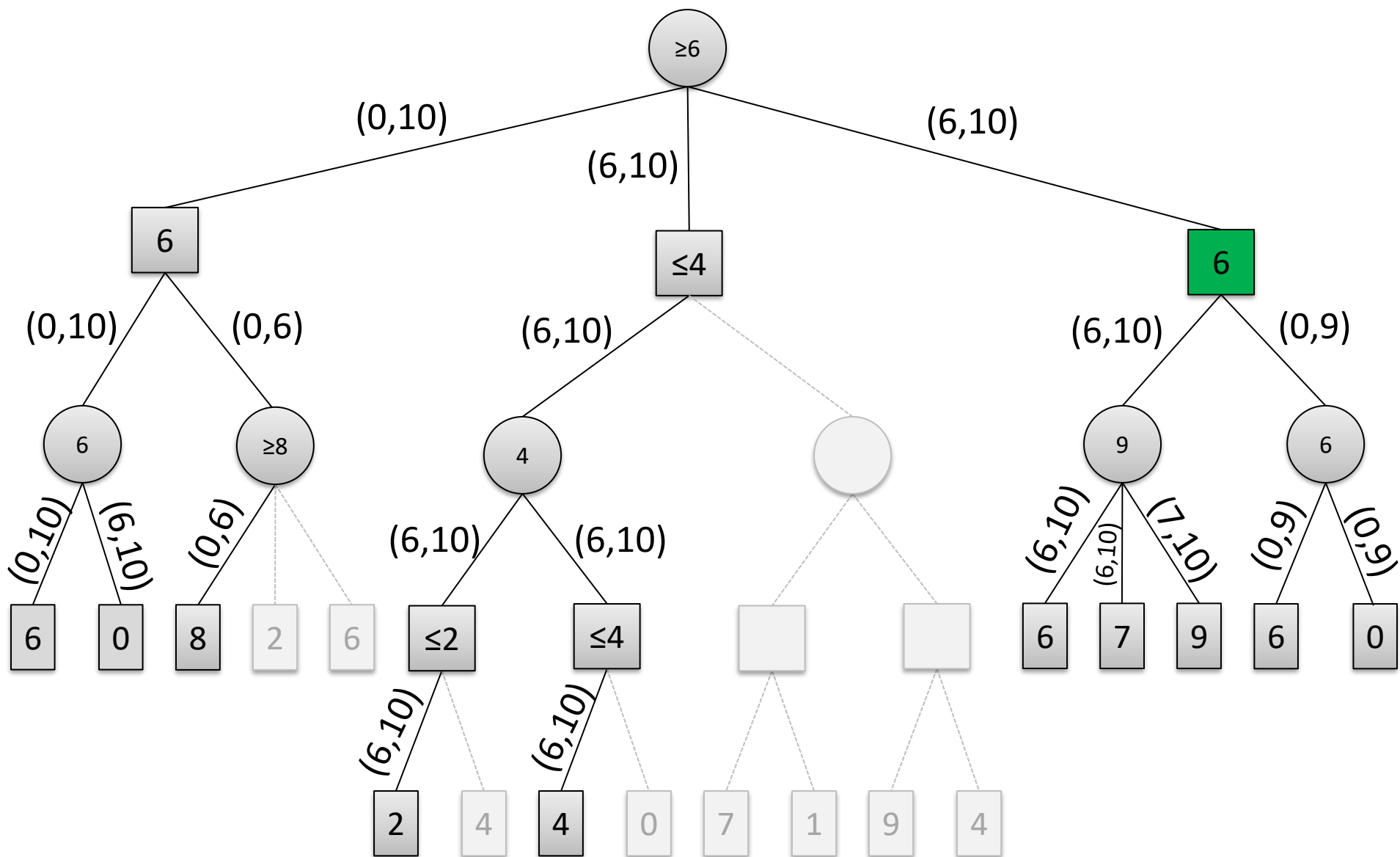
Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max

 ruch gracza Min



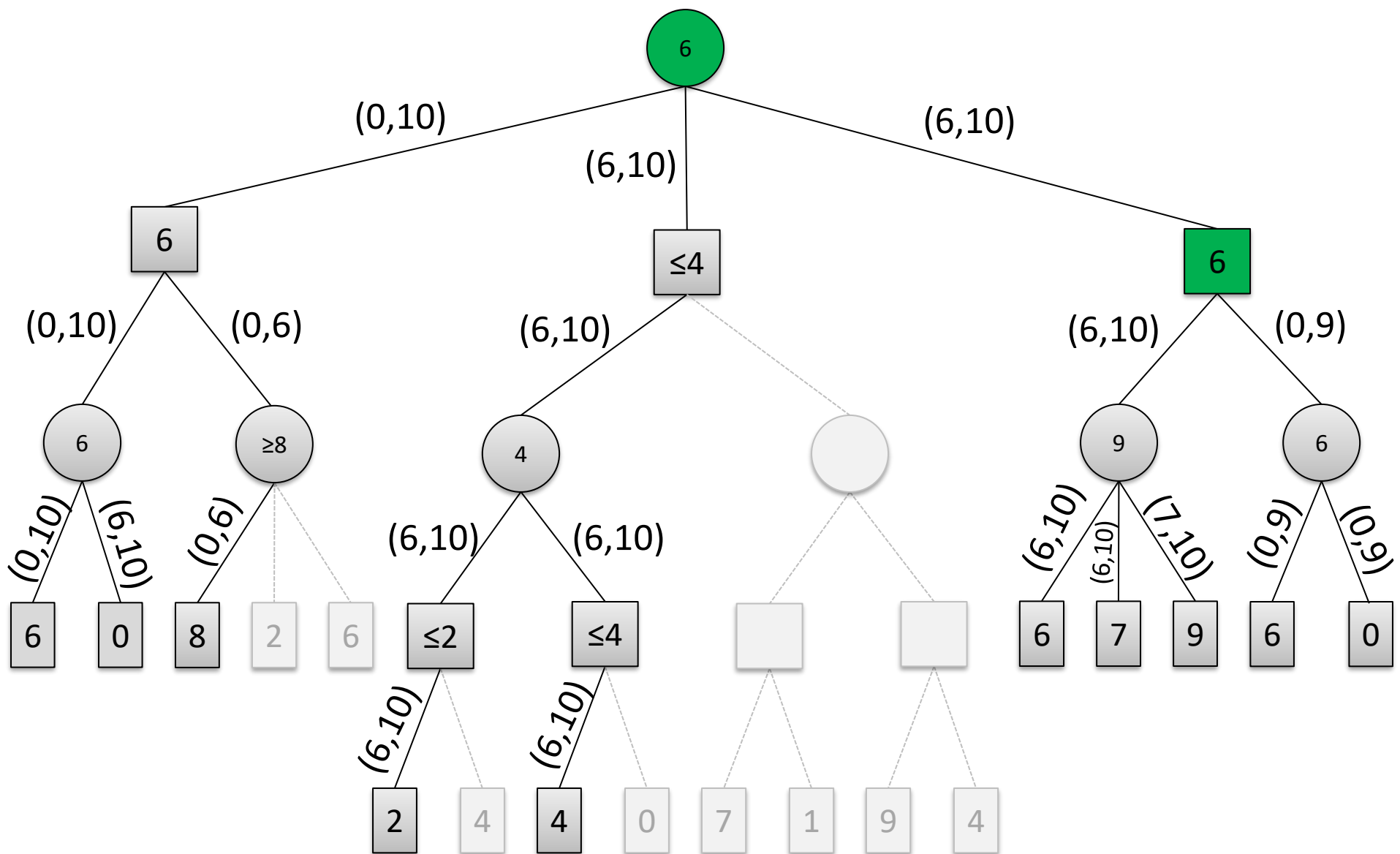
Legenda:



ruch gracza Max



ruch gracza Min



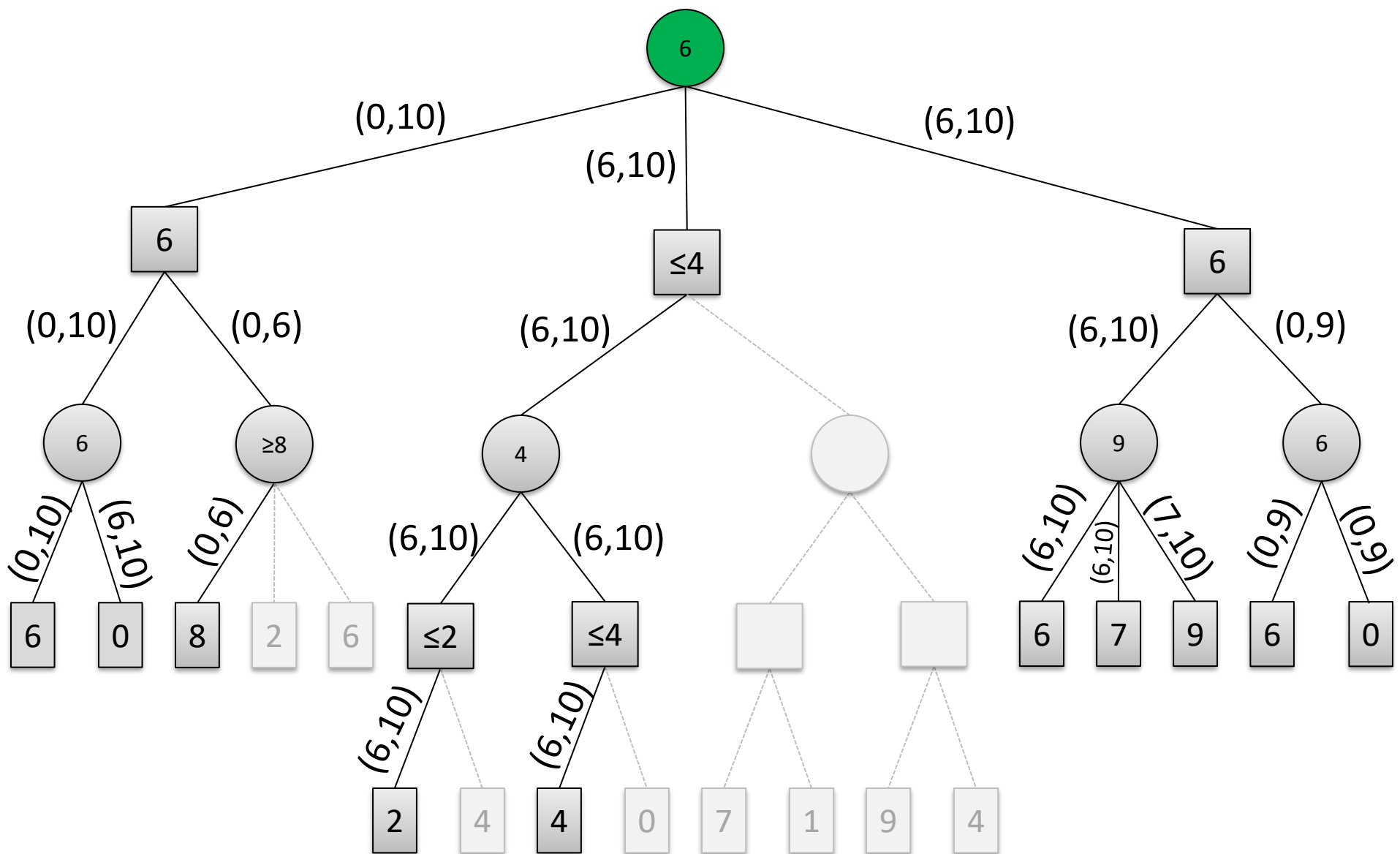
Legenda:



ruch gracza Max

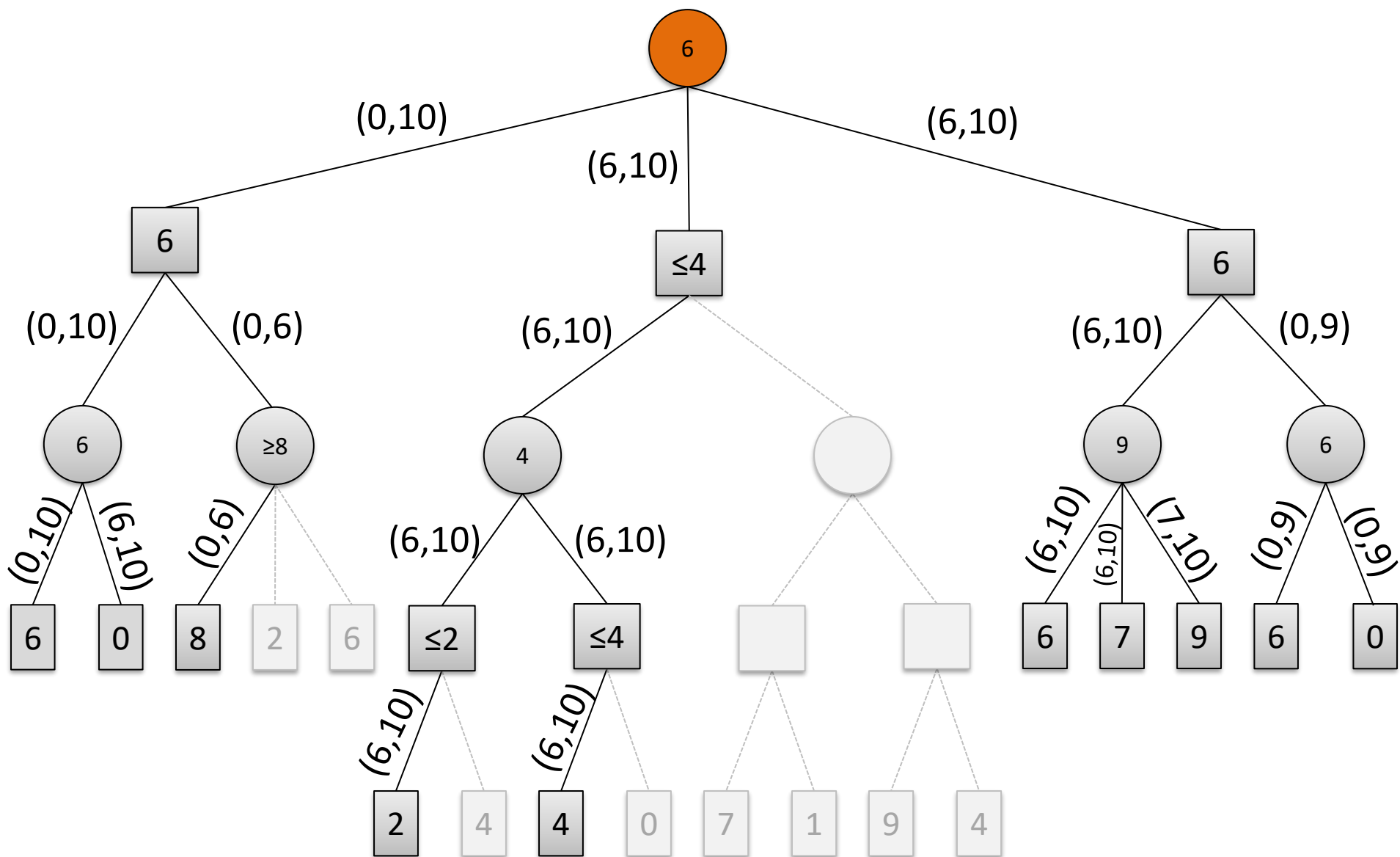


ruch gracza Min



Legenda:  ruch gracza Max

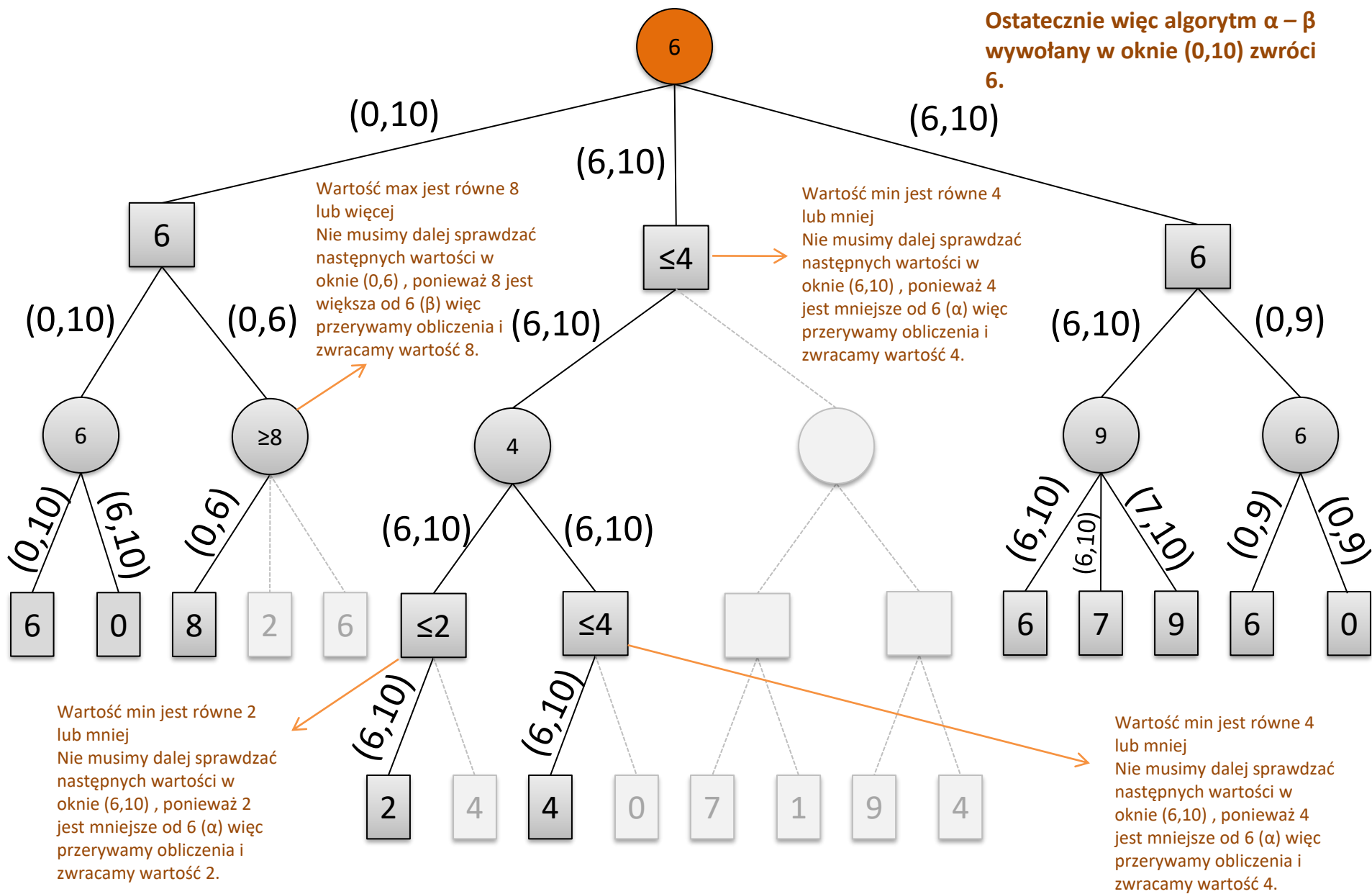
 ruch gracza Min



Legenda:  ruch gracza Max

 ruch gracza Min

Ostatecznie więc algorytm $\alpha - \beta$ wywołany w oknie (0,10) zwróci 6.



Legenda:



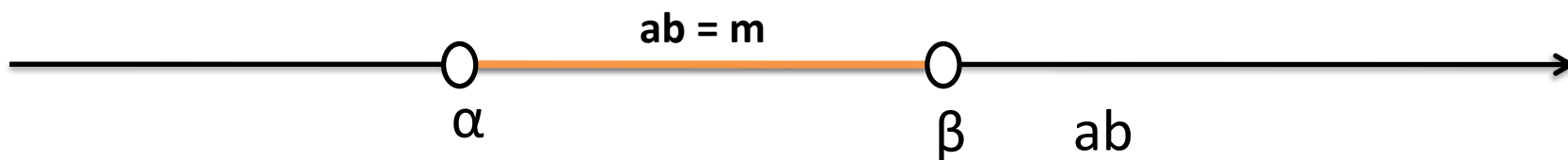
ruch gracza Max



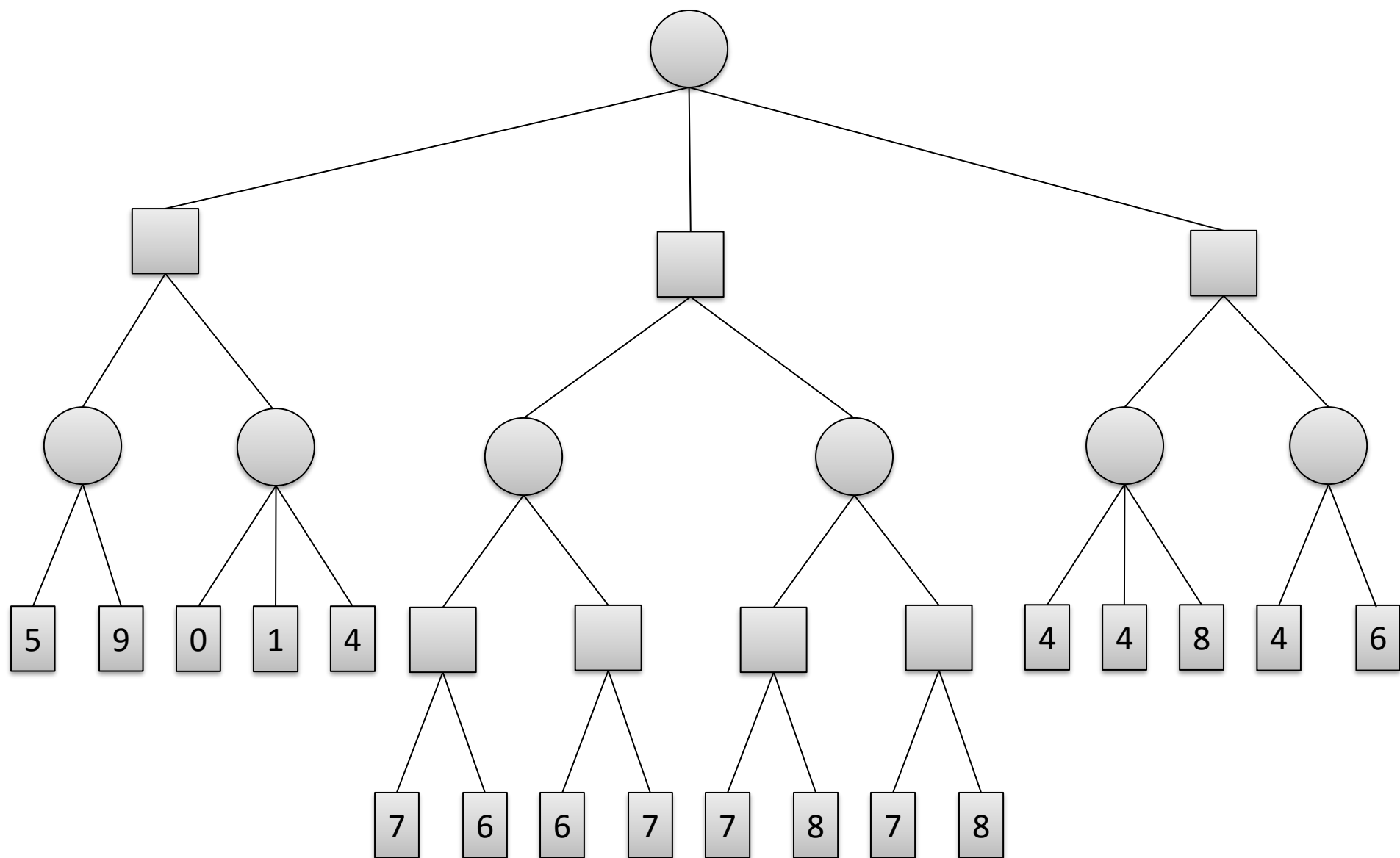
ruch gracza Min

Niech (S, s_1, N, ev) – gra, $s \in S$ – stan gry, $0 \leq \alpha < \beta \leq 1$.
Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz
 $m = \text{MinMax}(s)$.

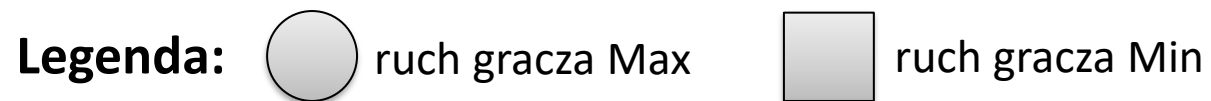
Algorytm Alfa – Beta zwraca wartość z przedziału α, β .
 $\alpha < ab < \beta \Rightarrow ab = m$

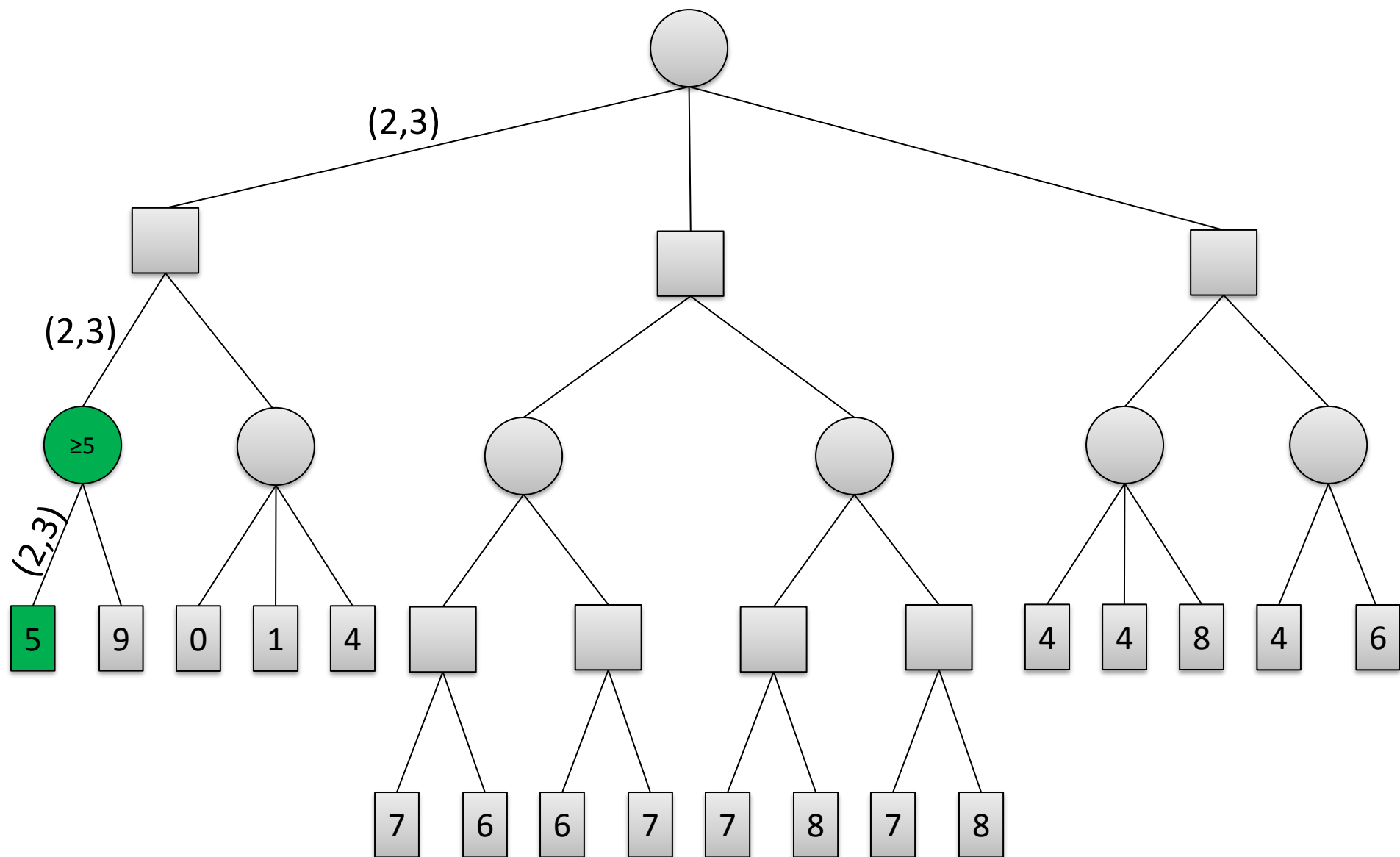


Przykład 3. Rozważmy działanie wywołanego algorytmu α - β w oknie (2,3) poniższego grafu gry:

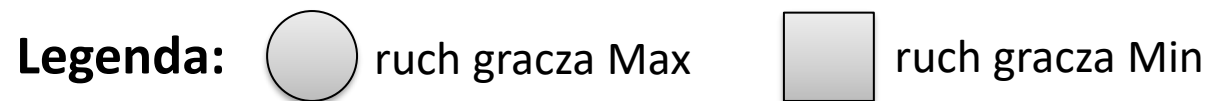


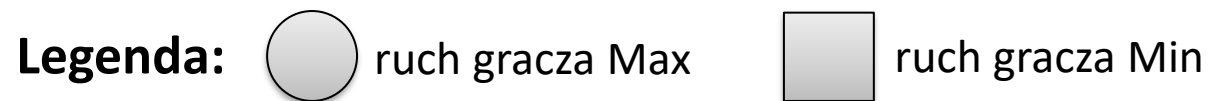


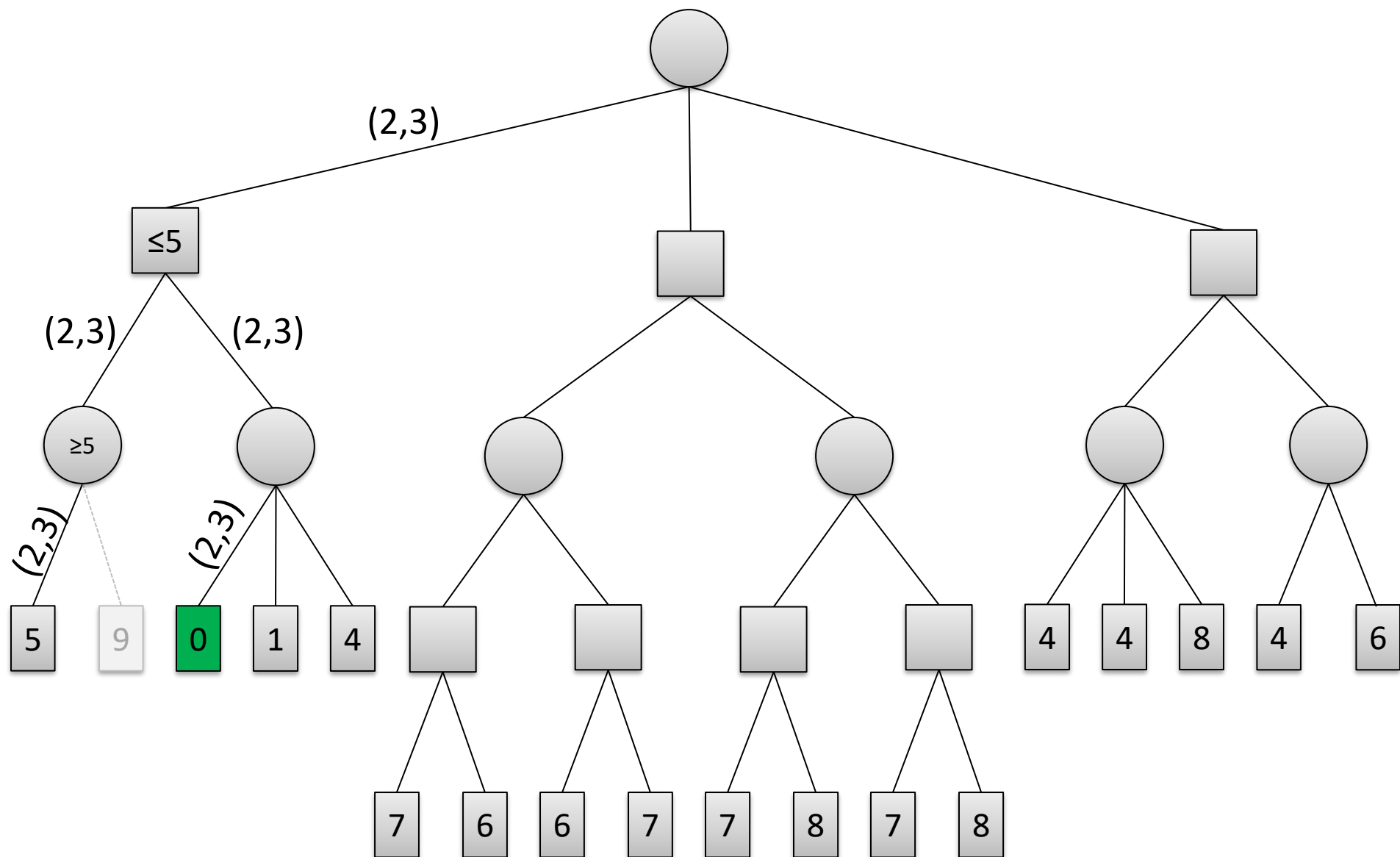




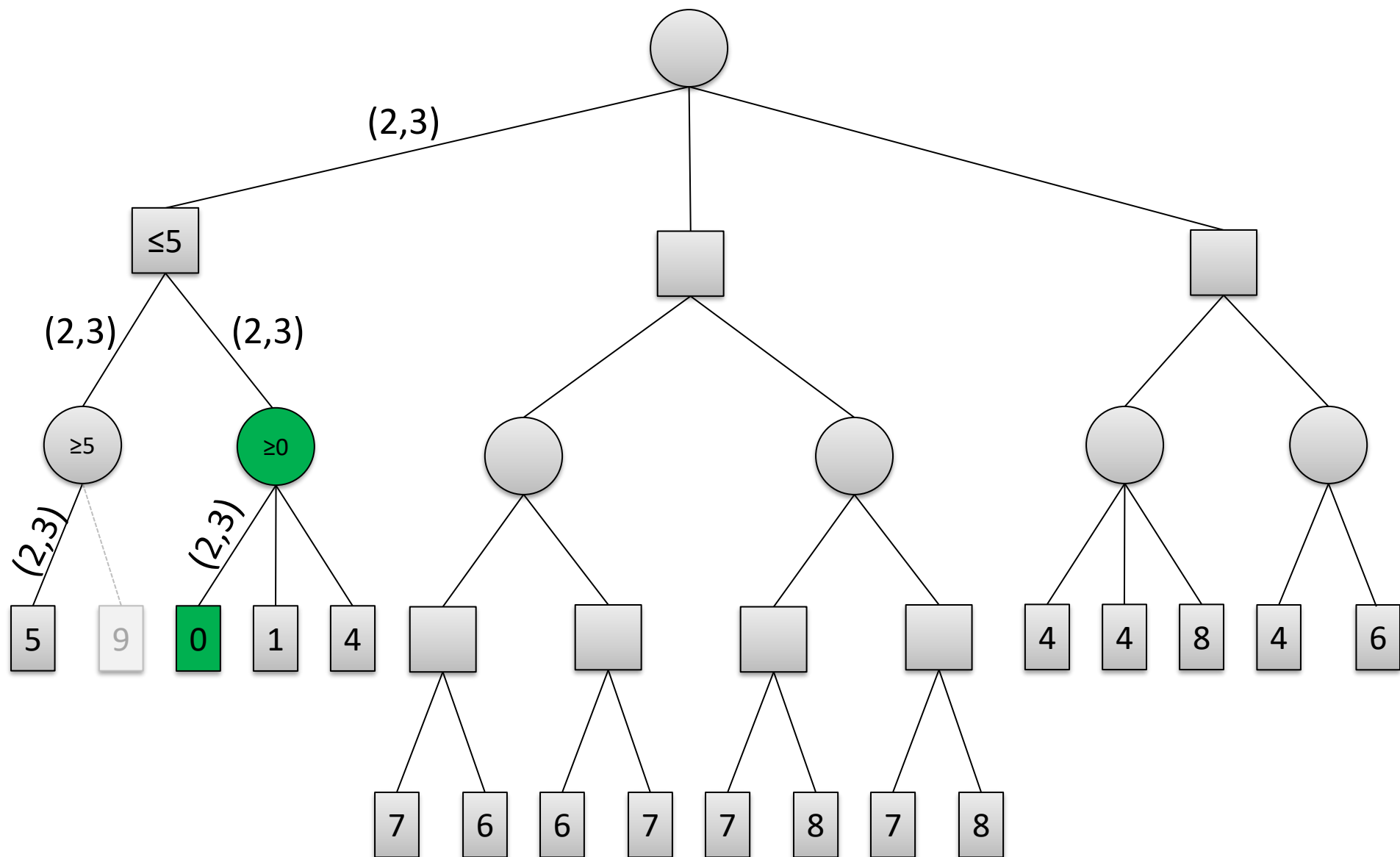
Legenda:  ruch gracza Max  ruch gracza Min





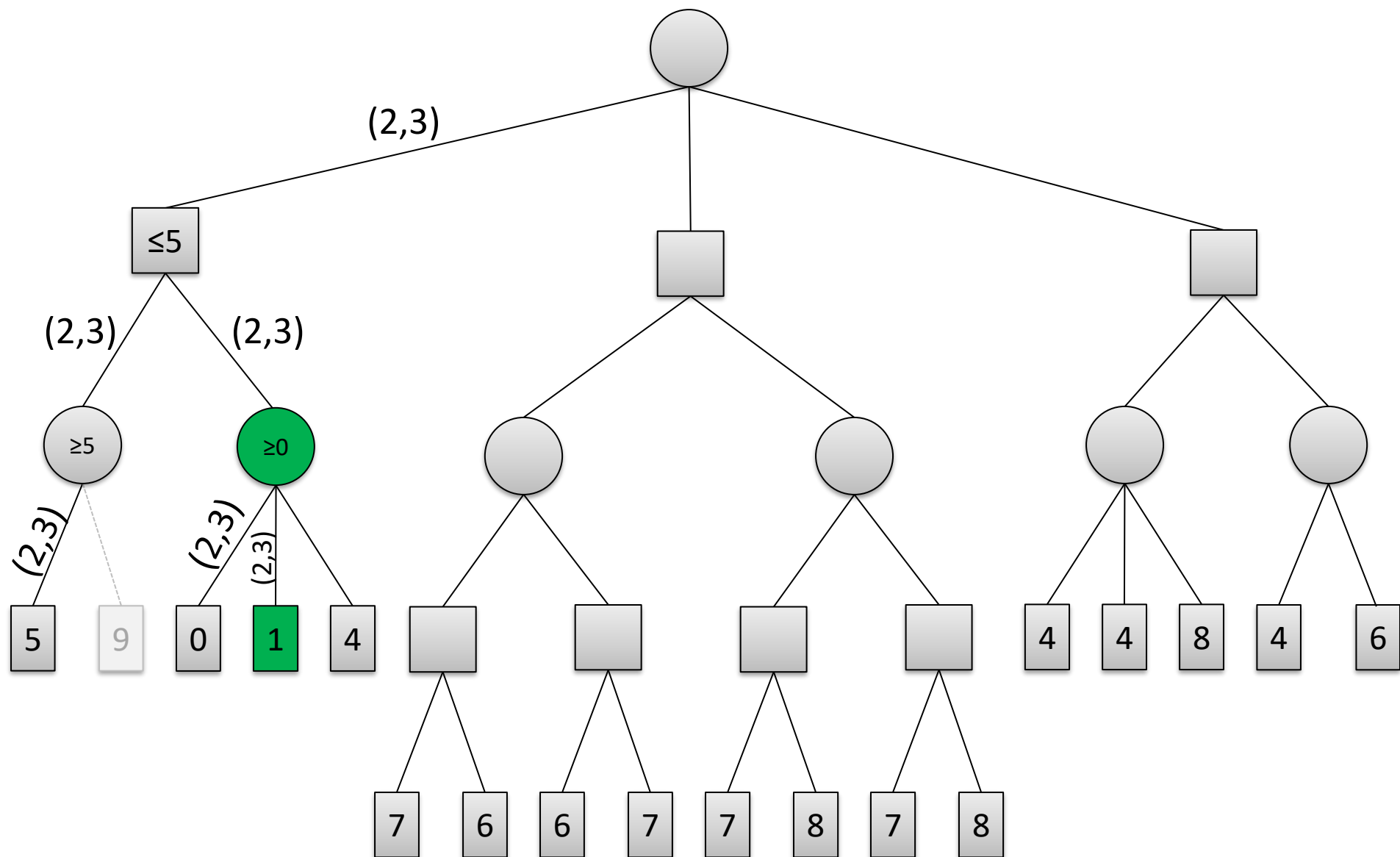


Legenda:  ruch gracza Max  ruch gracza Min



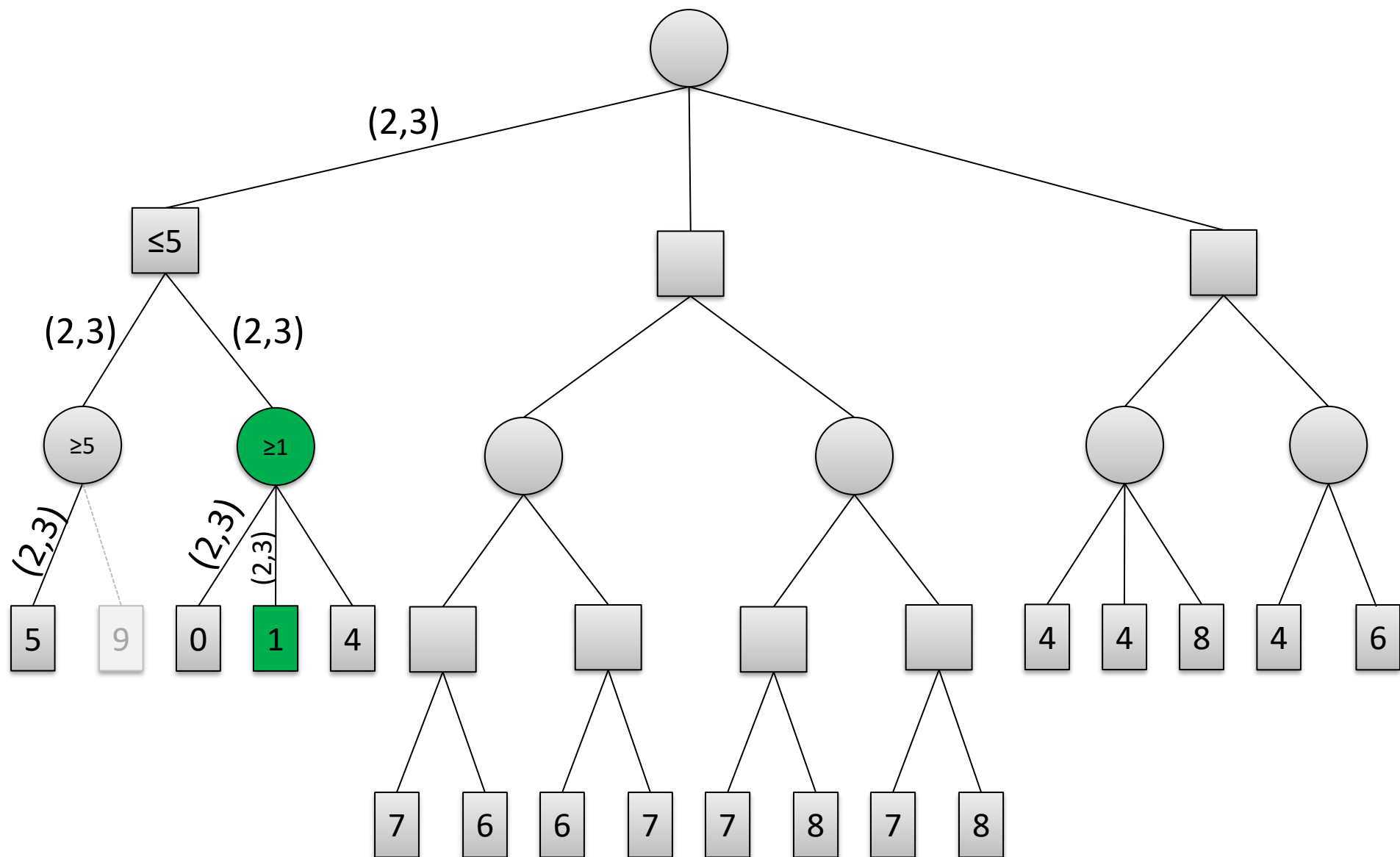
Legenda:  ruch gracza Max

 ruch gracza Min

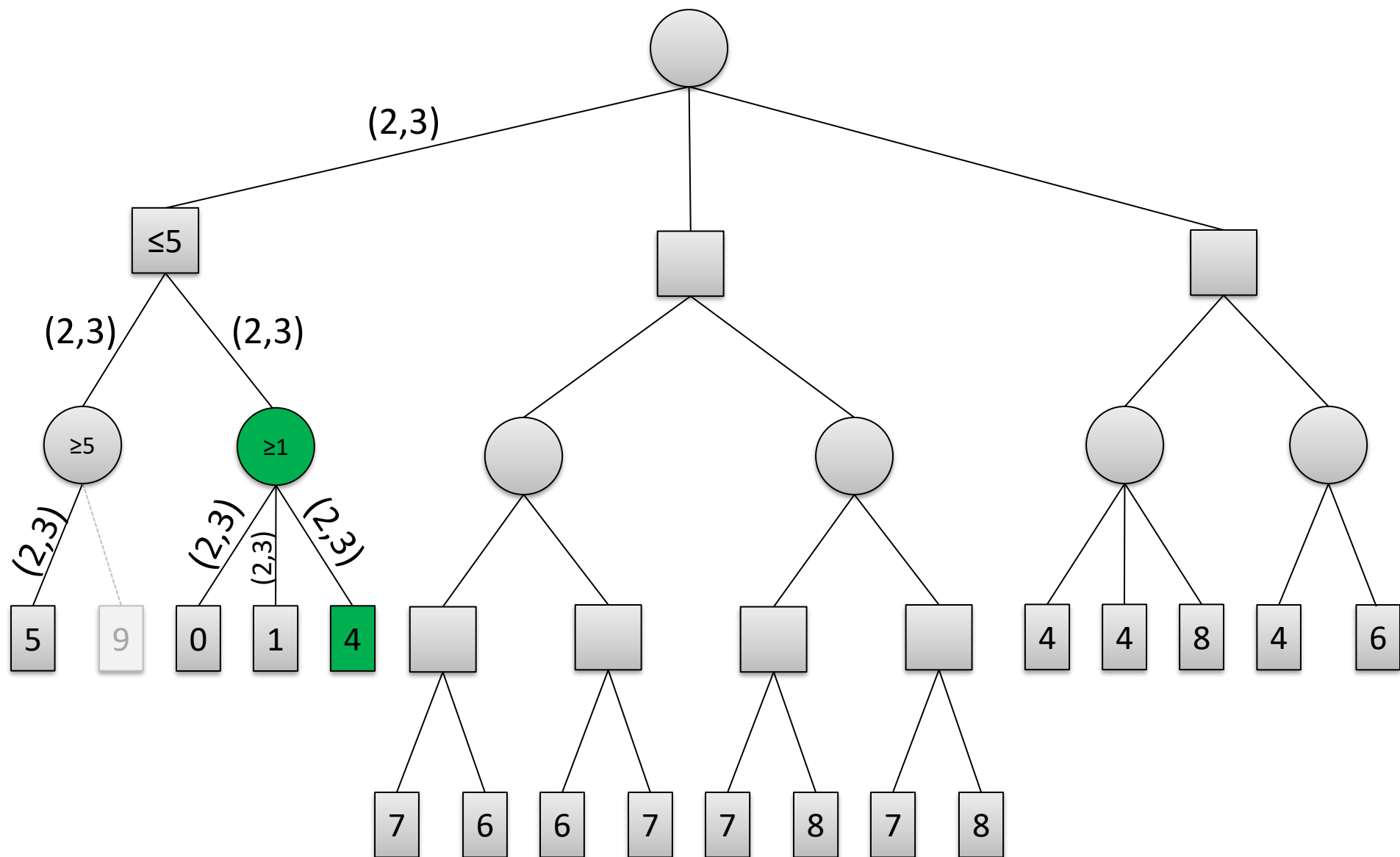


Legenda:  ruch gracza Max

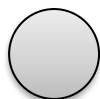
 ruch gracza Min



Legenda:  ruch gracza Max  ruch gracza Min



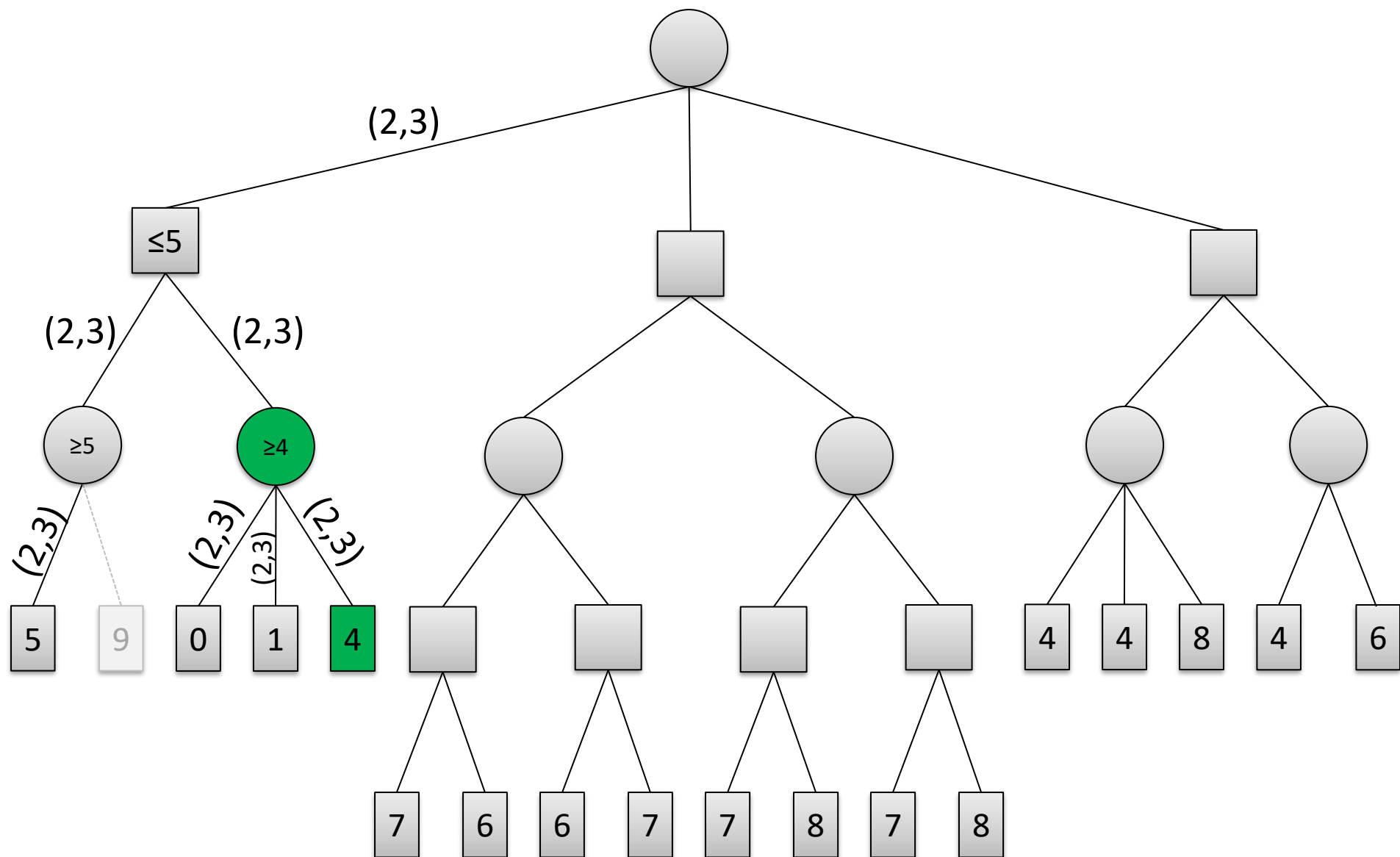
Legenda:



ruch gracza Max

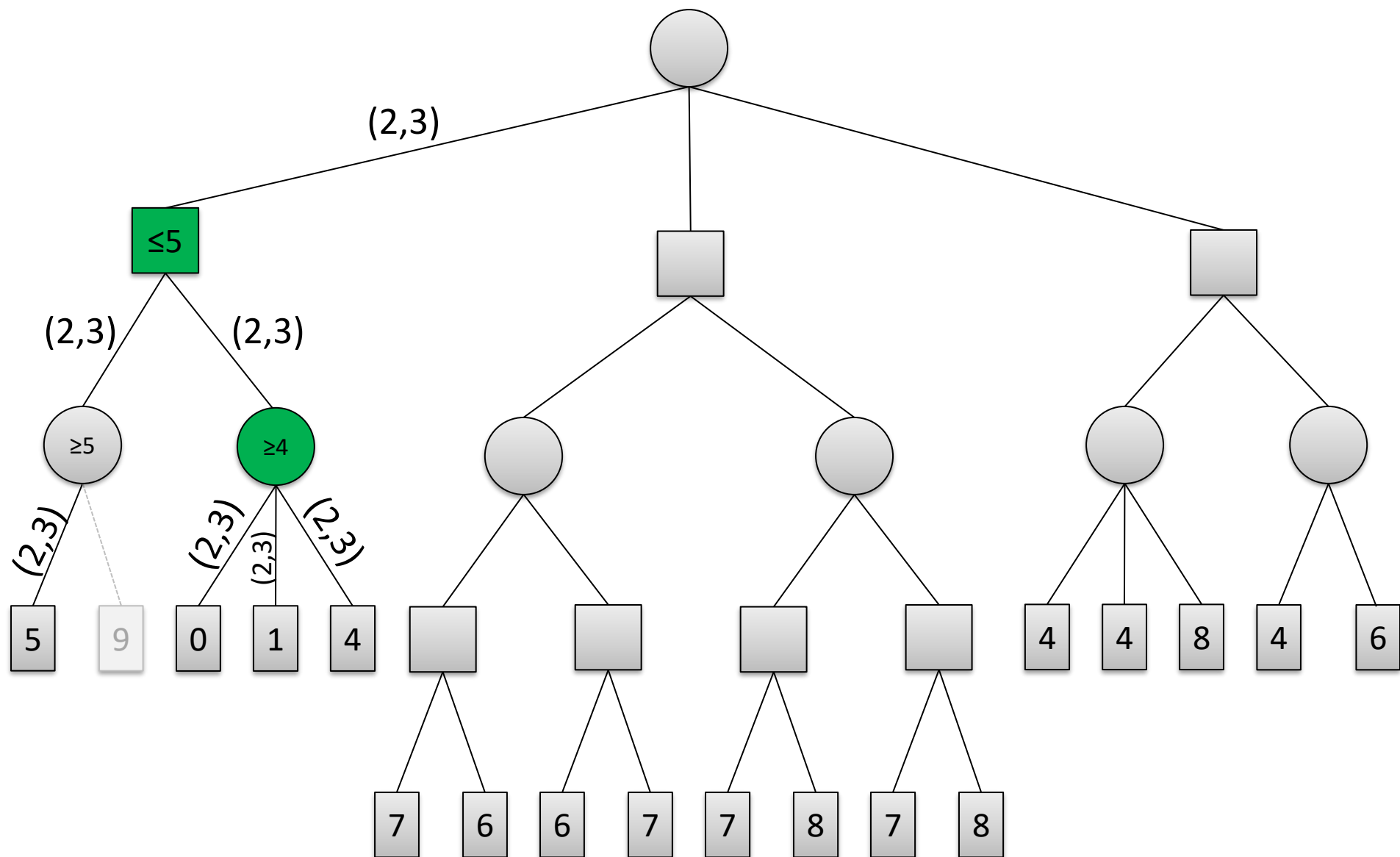


ruch gracza Min



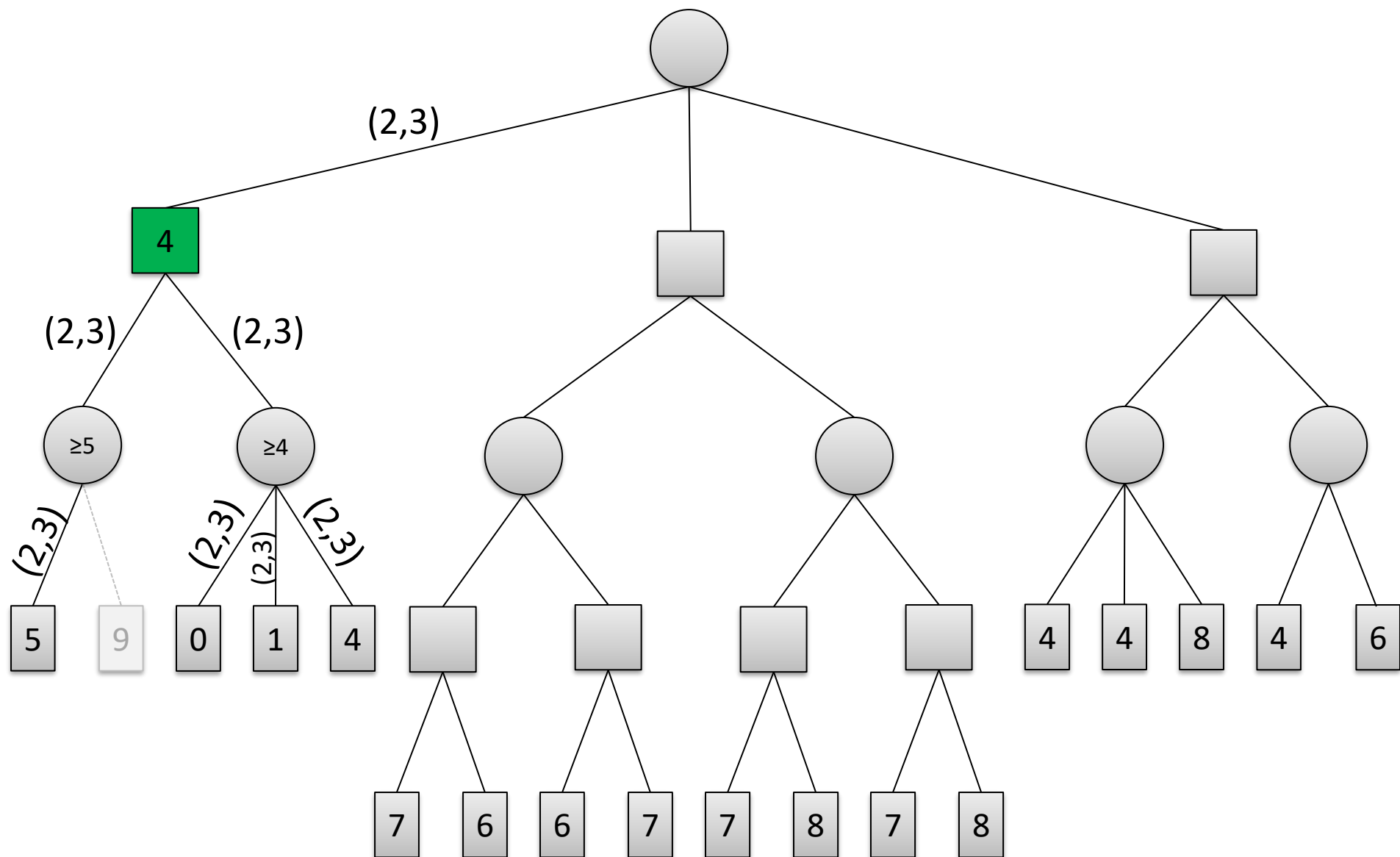
Legenda:  ruch gracza Max

 ruch gracza Min

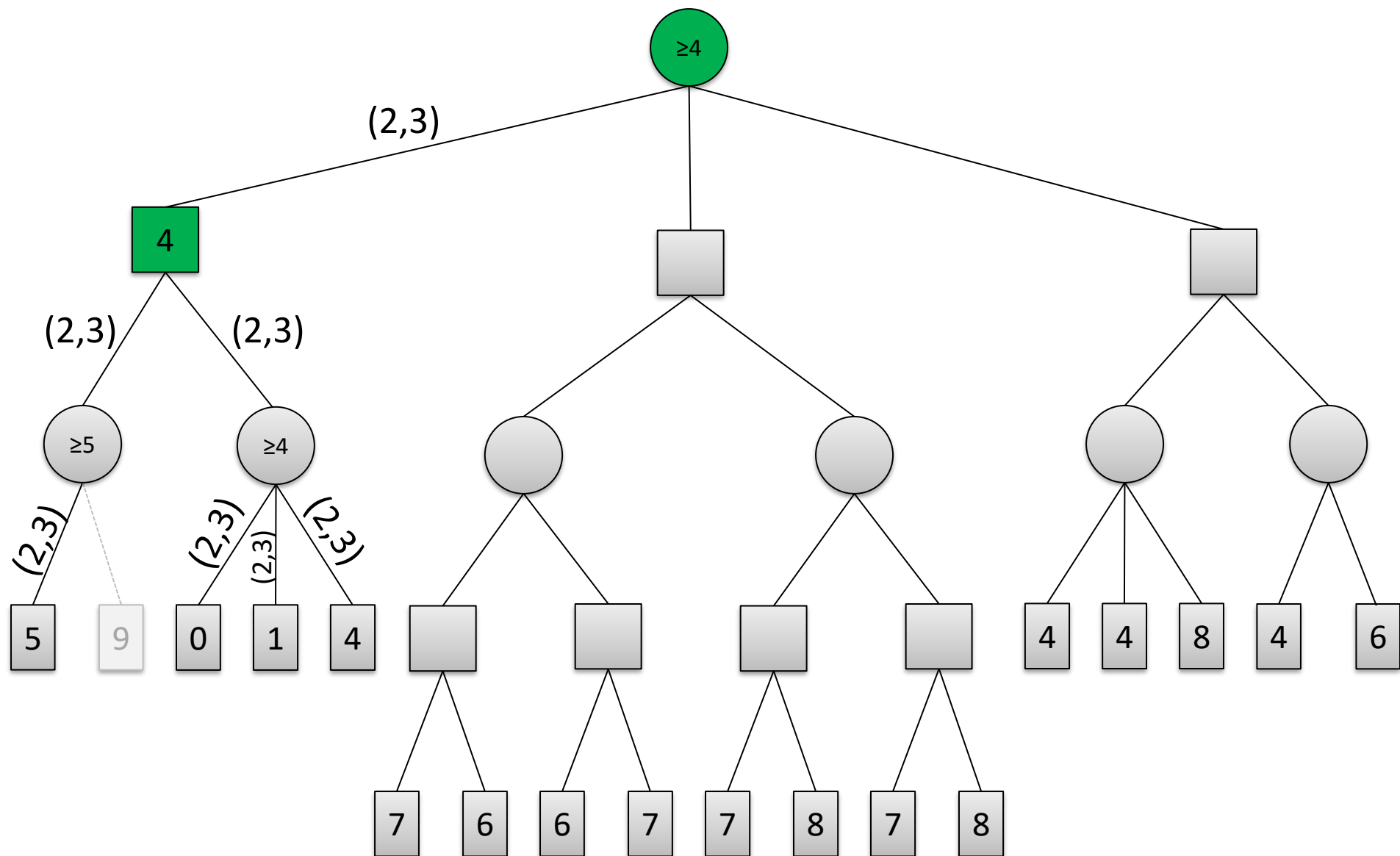


Legenda:  ruch gracza Max

 ruch gracza Min

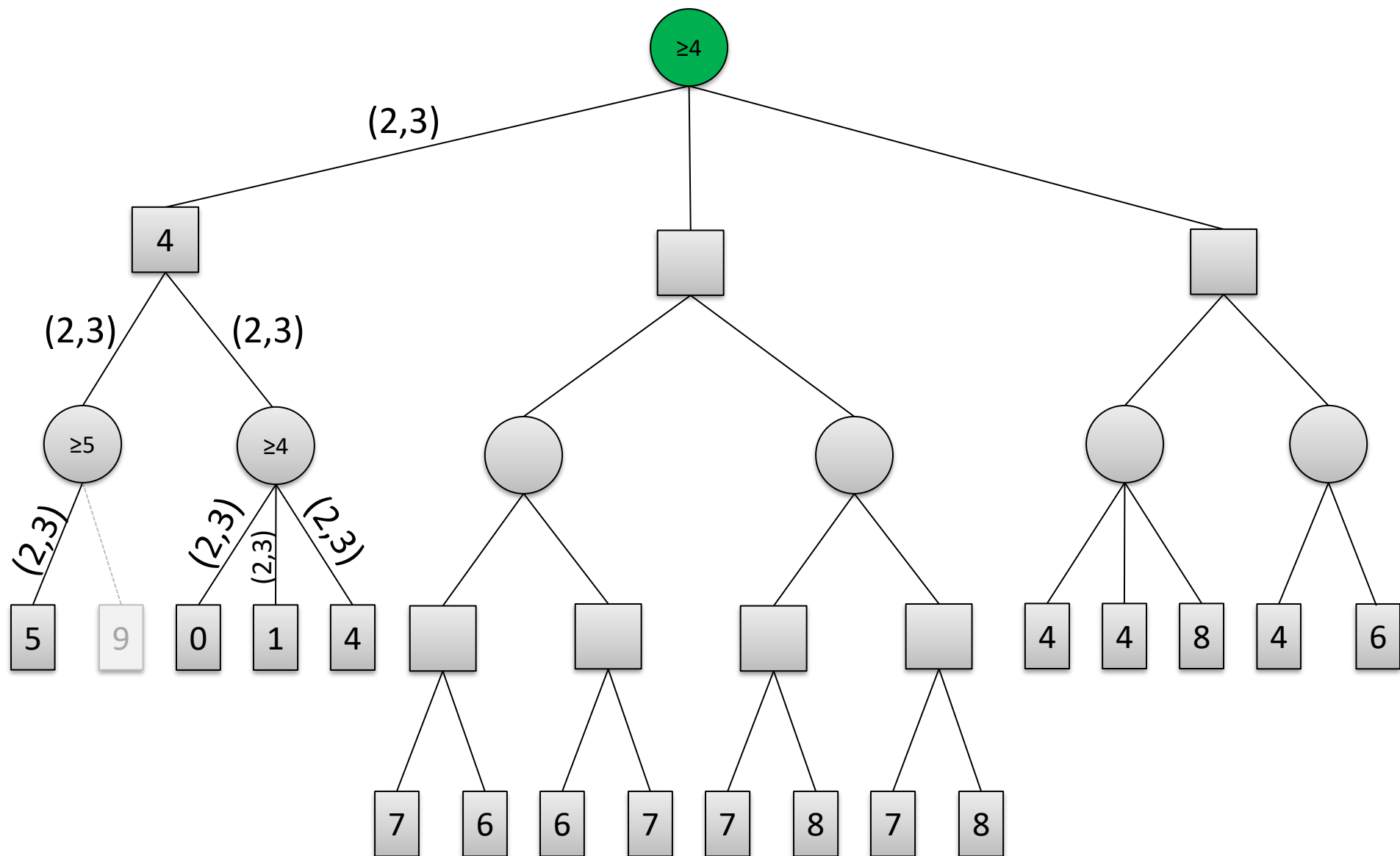


Legenda:  ruch gracza Max  ruch gracza Min

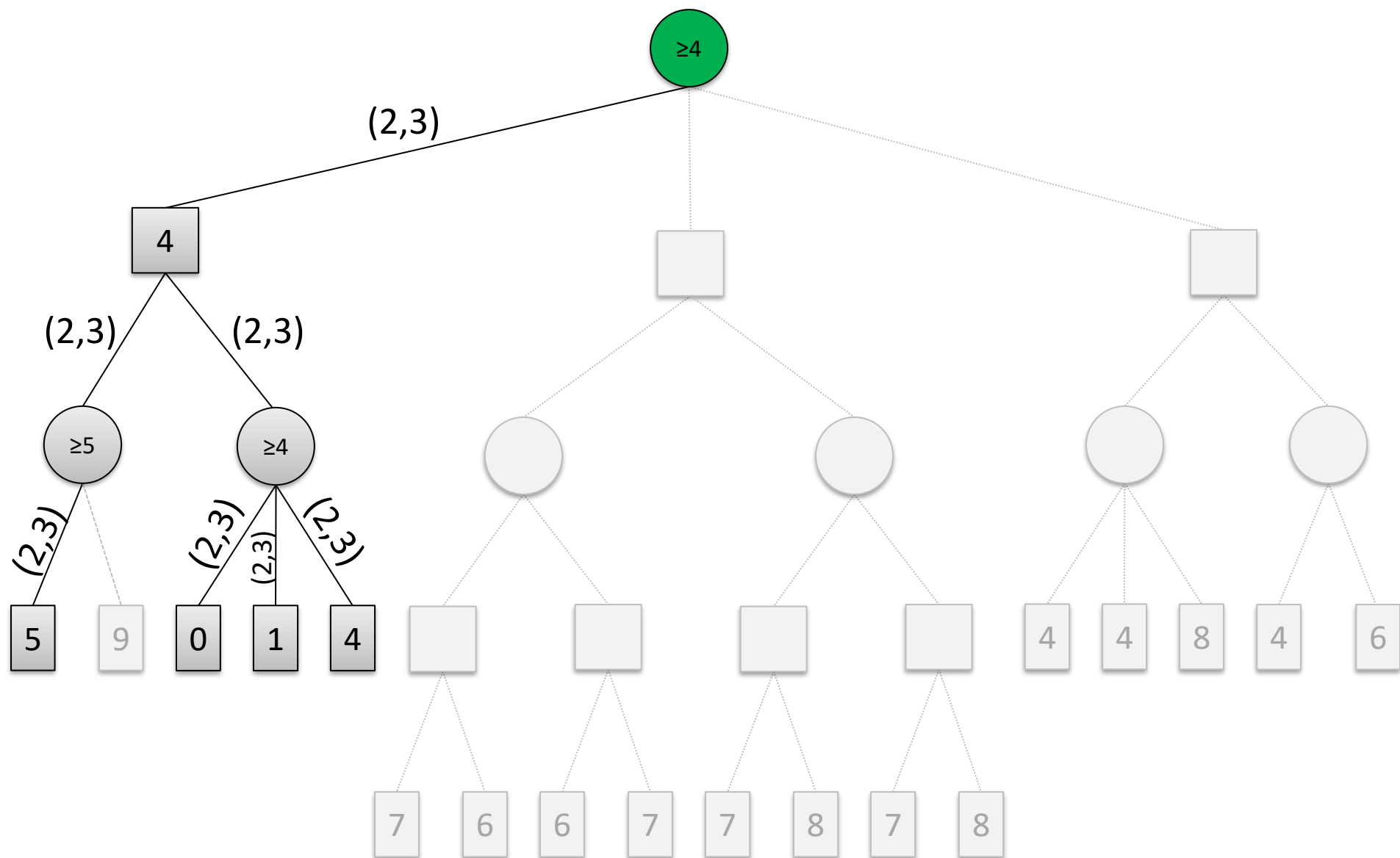


Legenda:  ruch gracza Max

 ruch gracza Min

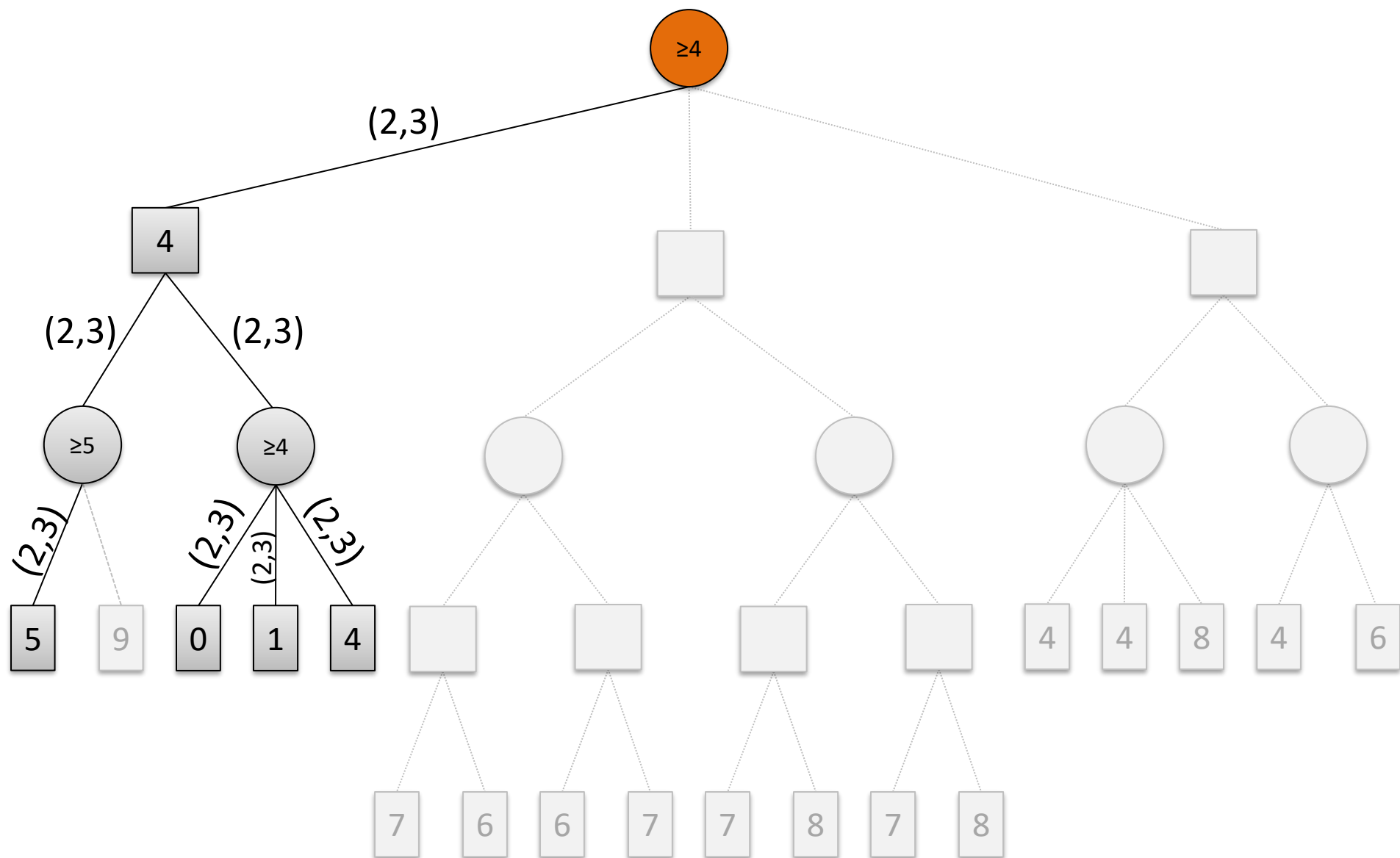


Legenda:  ruch gracza Max  ruch gracza Min



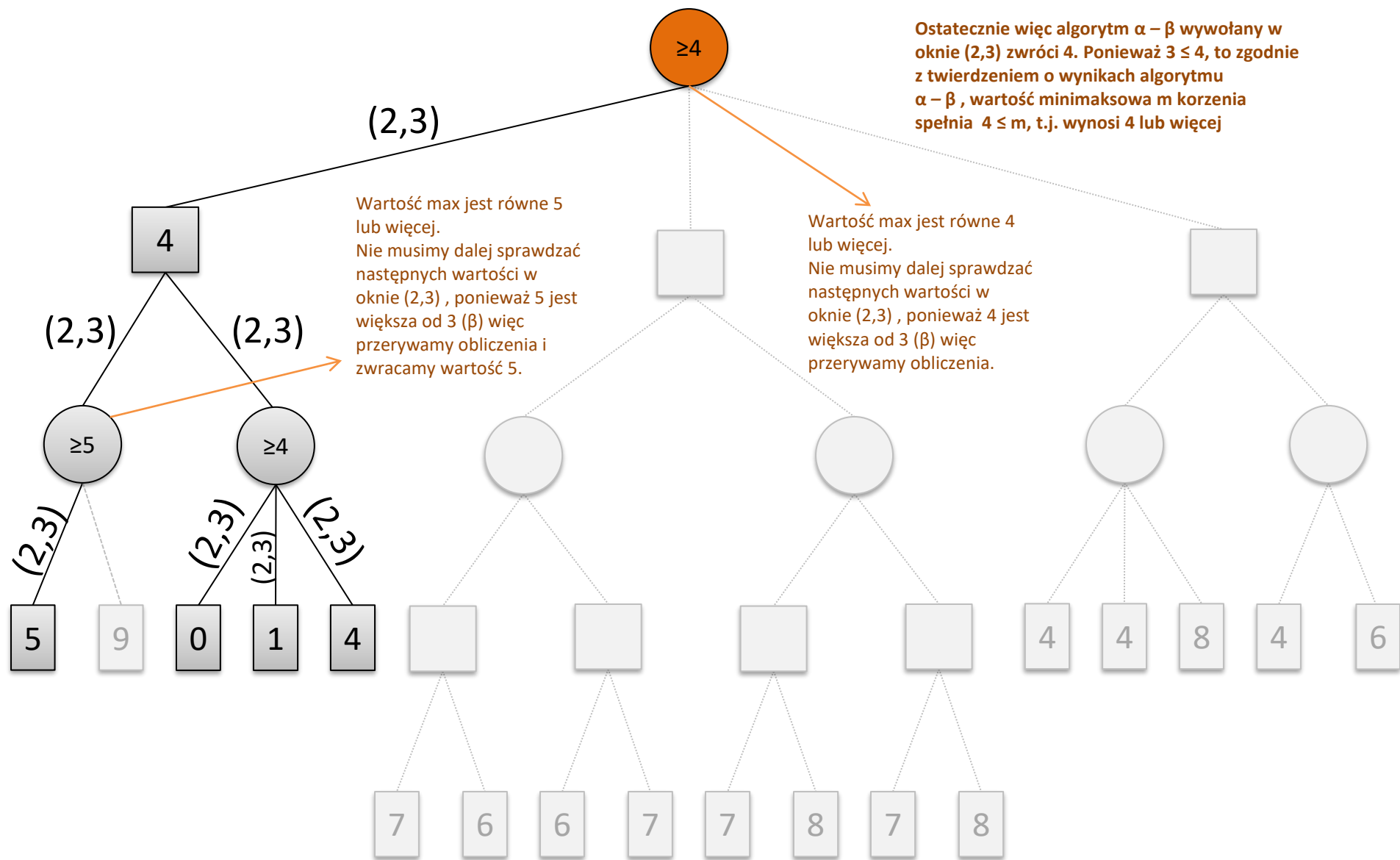
Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max

 ruch gracza Min



Legenda:  ruch gracza Max  ruch gracza Min

Niech $(\mathbb{S}, s, N, ev)$ – gra, $s \in \mathbb{S}$ – stan gry, $0 \leq \alpha < \beta \leq 1$.
Oznaczmy ponadto $ab = \text{AlfaBeta}(s, \alpha, \beta)$ oraz
 $m = \text{MinMax}(s)$.

Algorytm Alfa – Beta zwraca wartość powyżej β .
 $\beta \leq ab \Rightarrow ab \leq m$



Literatura

- [1] Algorytm alfa-beta. Dostępne na stronie:
https://pl.wikipedia.org/wiki/Algorytm_alfa-beta
- [2] Notatki z zajęć. Teoria gier i kombinatoryka, Uniwersytet Łódzki, semestr letni 2019/2020.
- [3] D.Knuth and R.More. An analysis of alpha-beta pruning. Artificial Intelligence 6, 1975.
- [4] Heurystyka. Dostępne na stronie:
[https://pl.wikipedia.org/wiki/Heurystyka_\(informatyka\)](https://pl.wikipedia.org/wiki/Heurystyka_(informatyka))
- [5] Metody heurystyczne. Dostępne na stronie:
<http://www.imio.polsl.pl/Dopobrania/MH%20MB4%20n2z%20w2.pdf>
- [6] Alpha Beta Pruning AI. Dostępne na stronie:
<https://www.professional-ai.com/alpha-beta-pruning.html>
- [7] G. Blokdyk, Alpha Beta Pruning. The Ultimate Step-By-Step Guide, 2017.
- [8] P. Beling, Algorytm α , β i MTD(f) – prezentacja , 2020.