



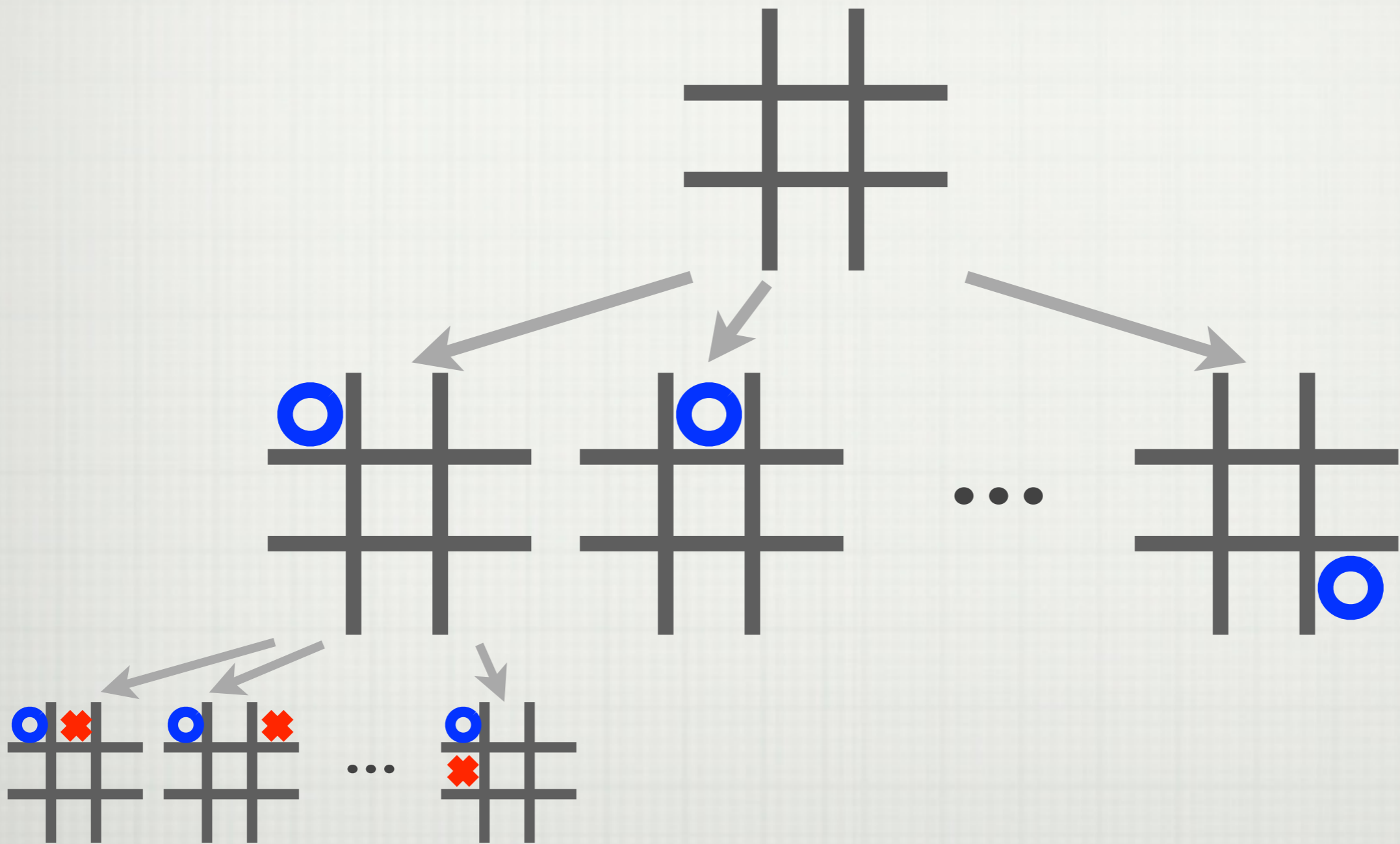
JOHDATUS TEKOÄLYYN

TEEMU ROOS

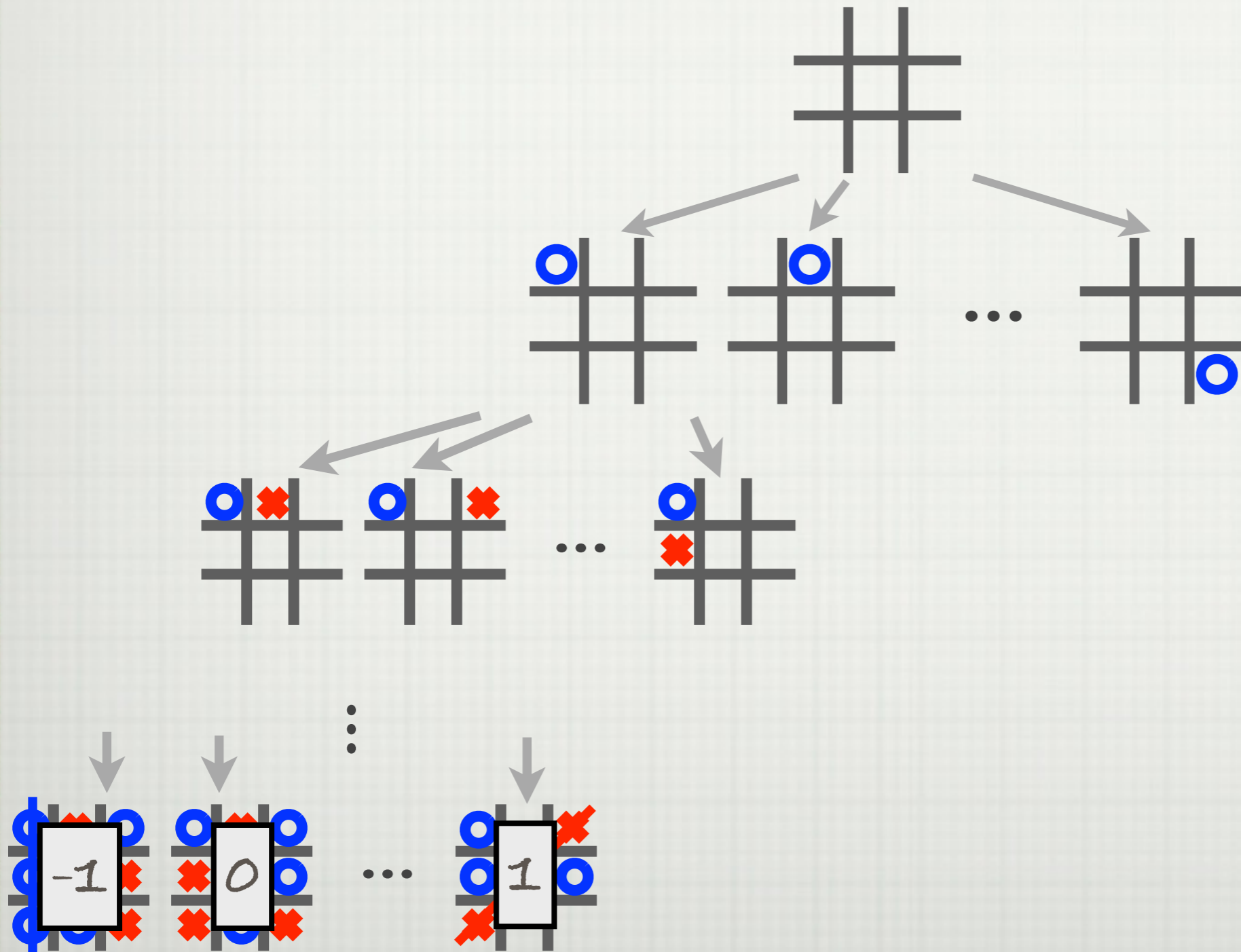


HELSINGIN YLIOPISTO

PELIPUU



PELIPUU



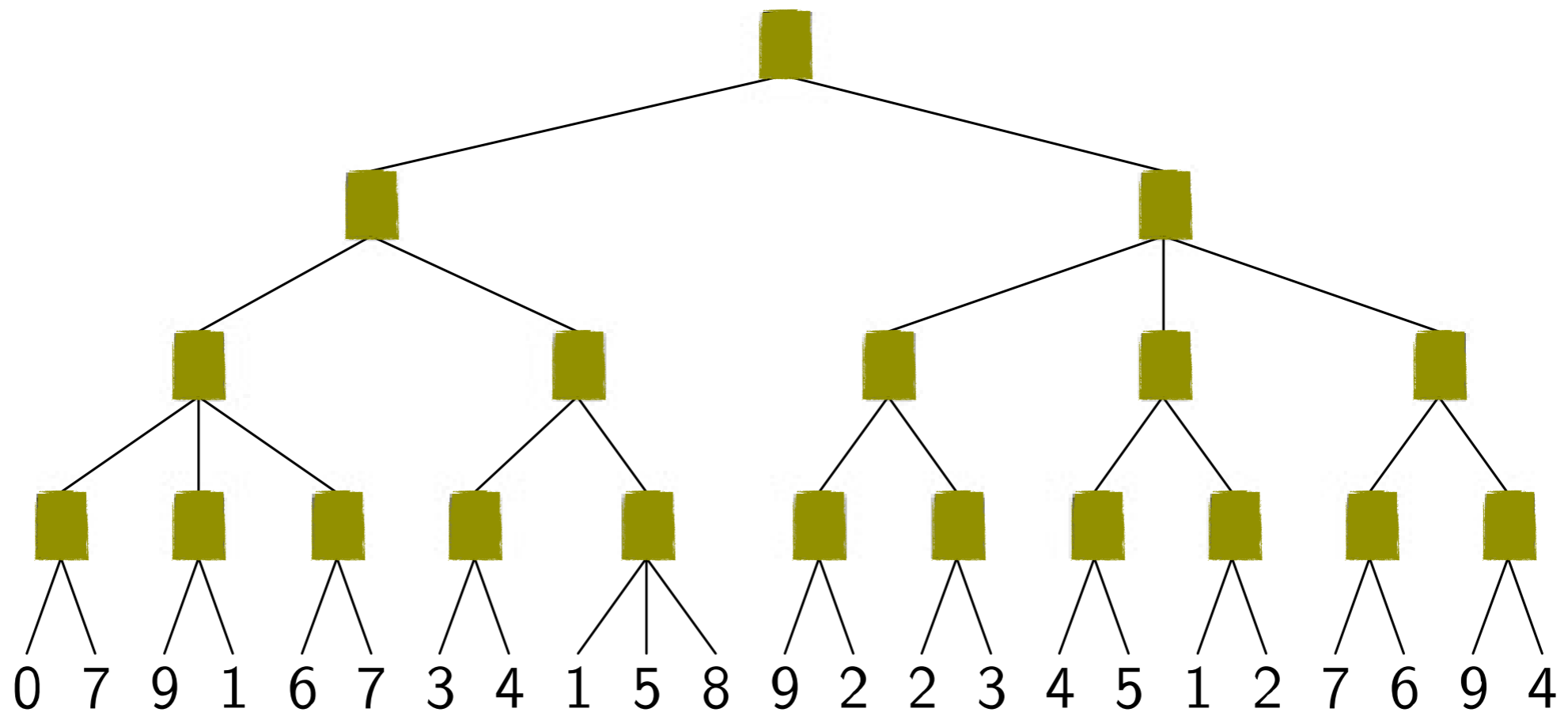
PELIPUU

MAX

MIN

MAX

MIN



PELIPUU

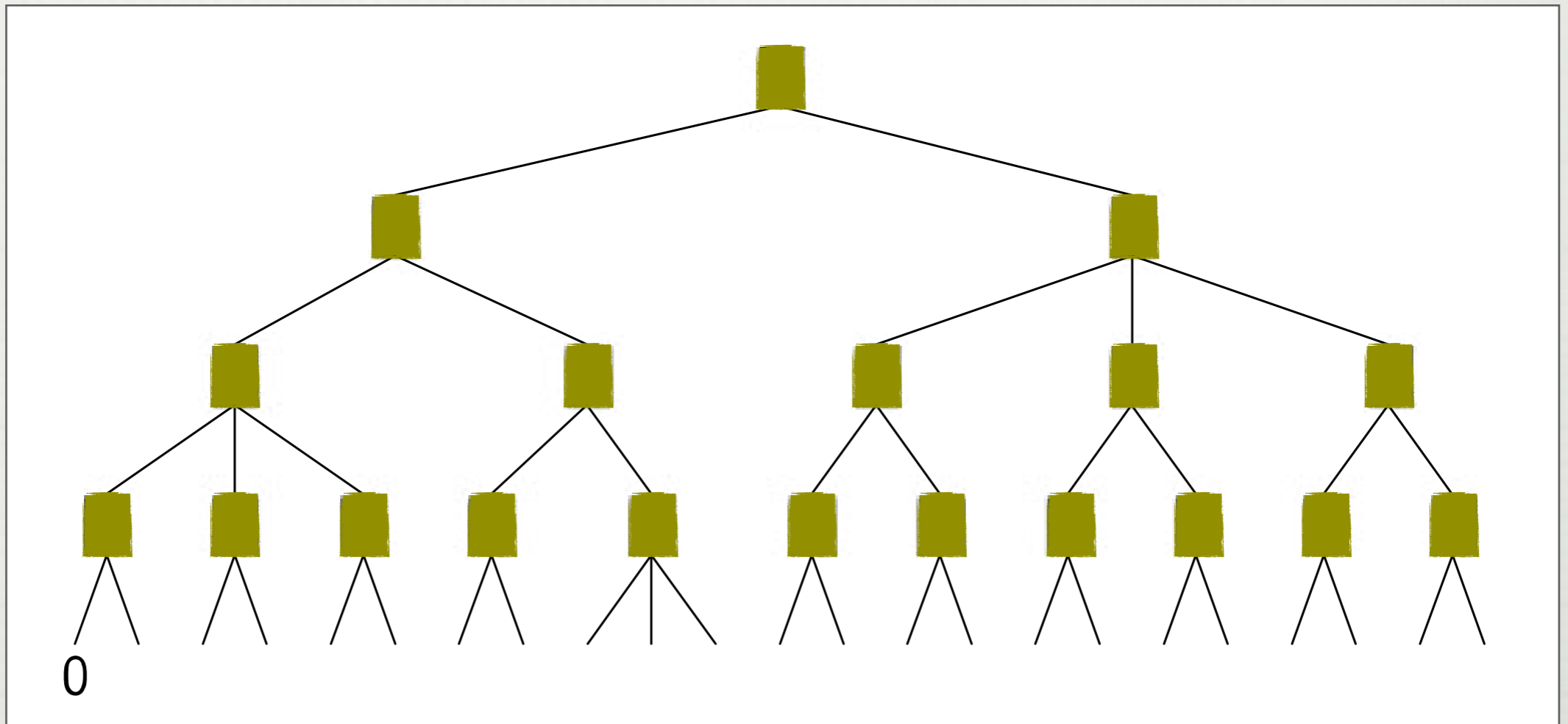
MAX

MIN

MAX

MIN

0



PELIPUU

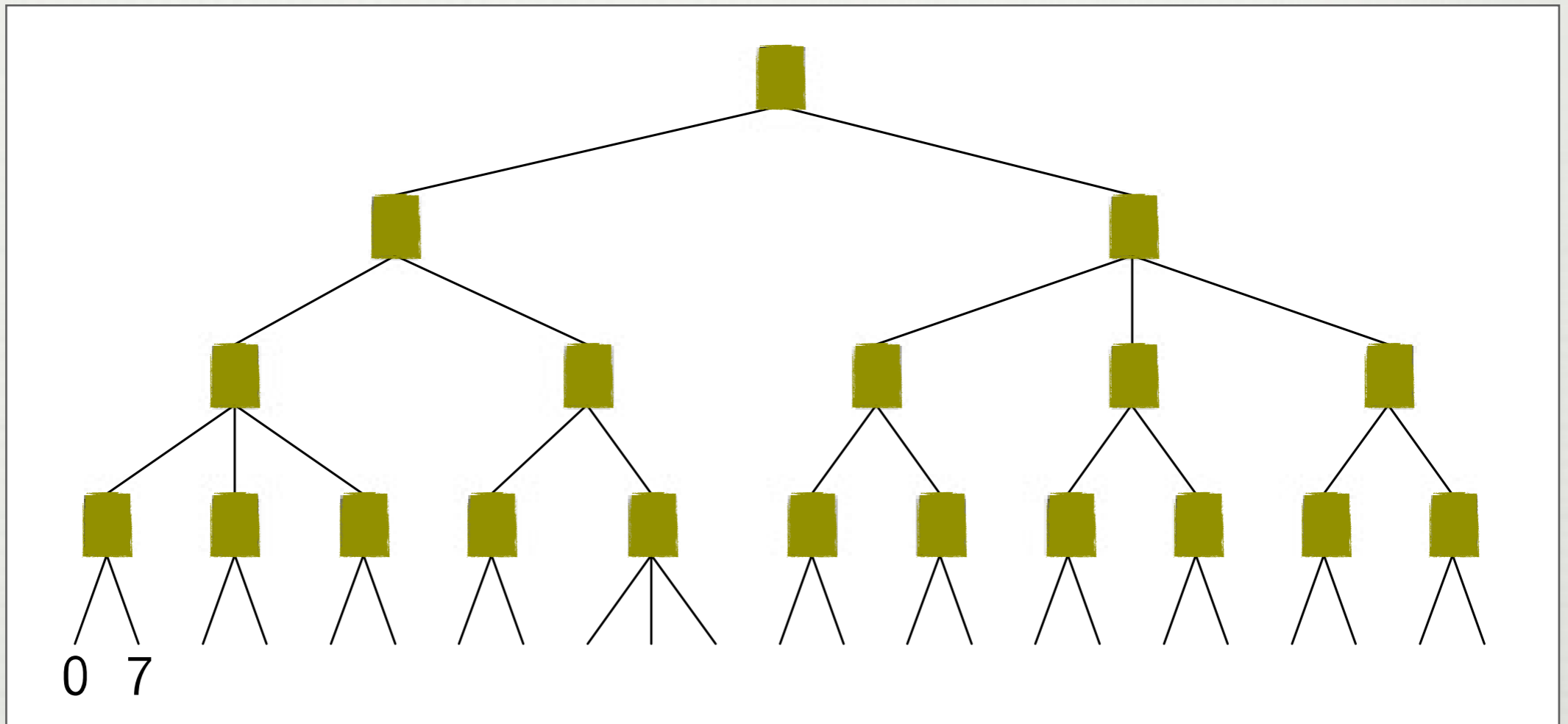
MAX

MIN

MAX

MIN

0 7



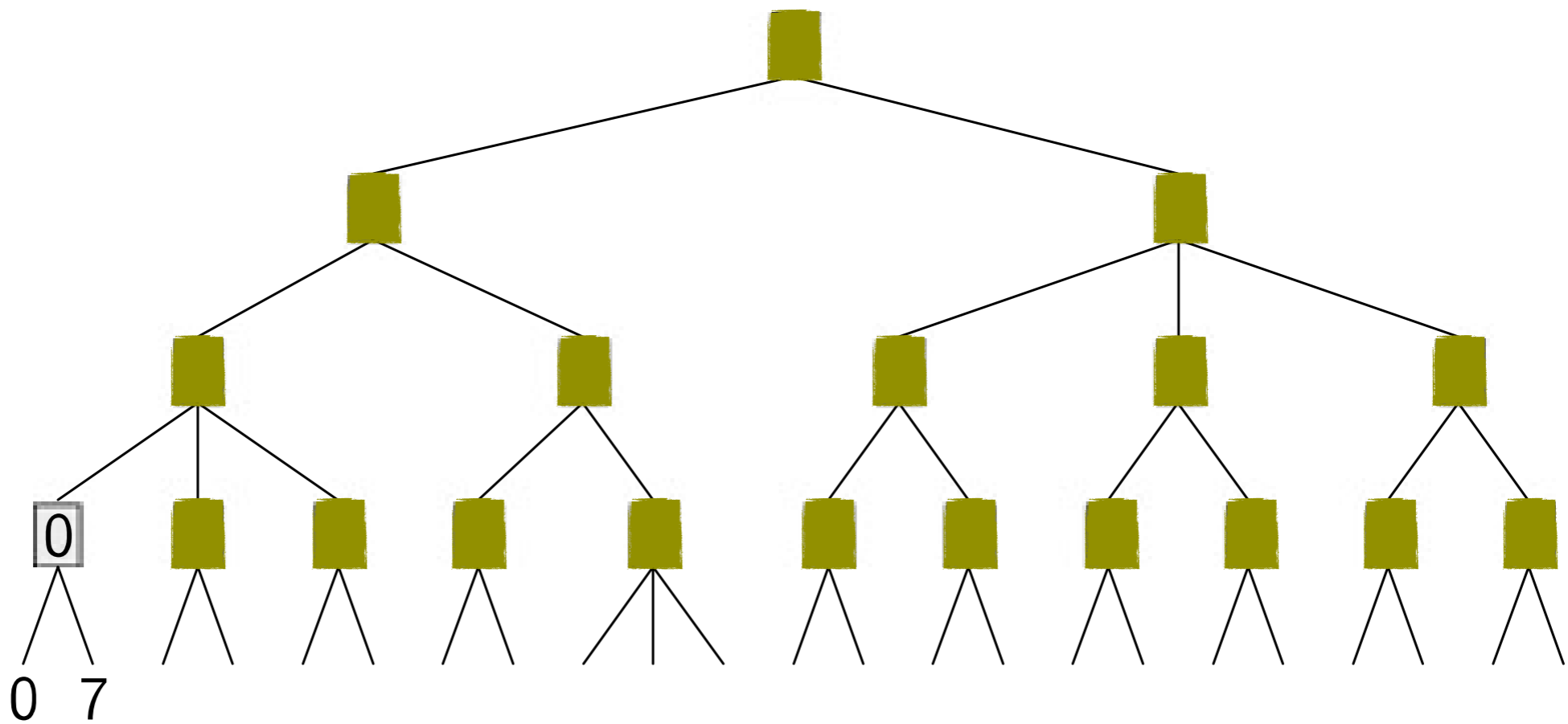
PELIPUU

MAX

MIN

MAX

MIN



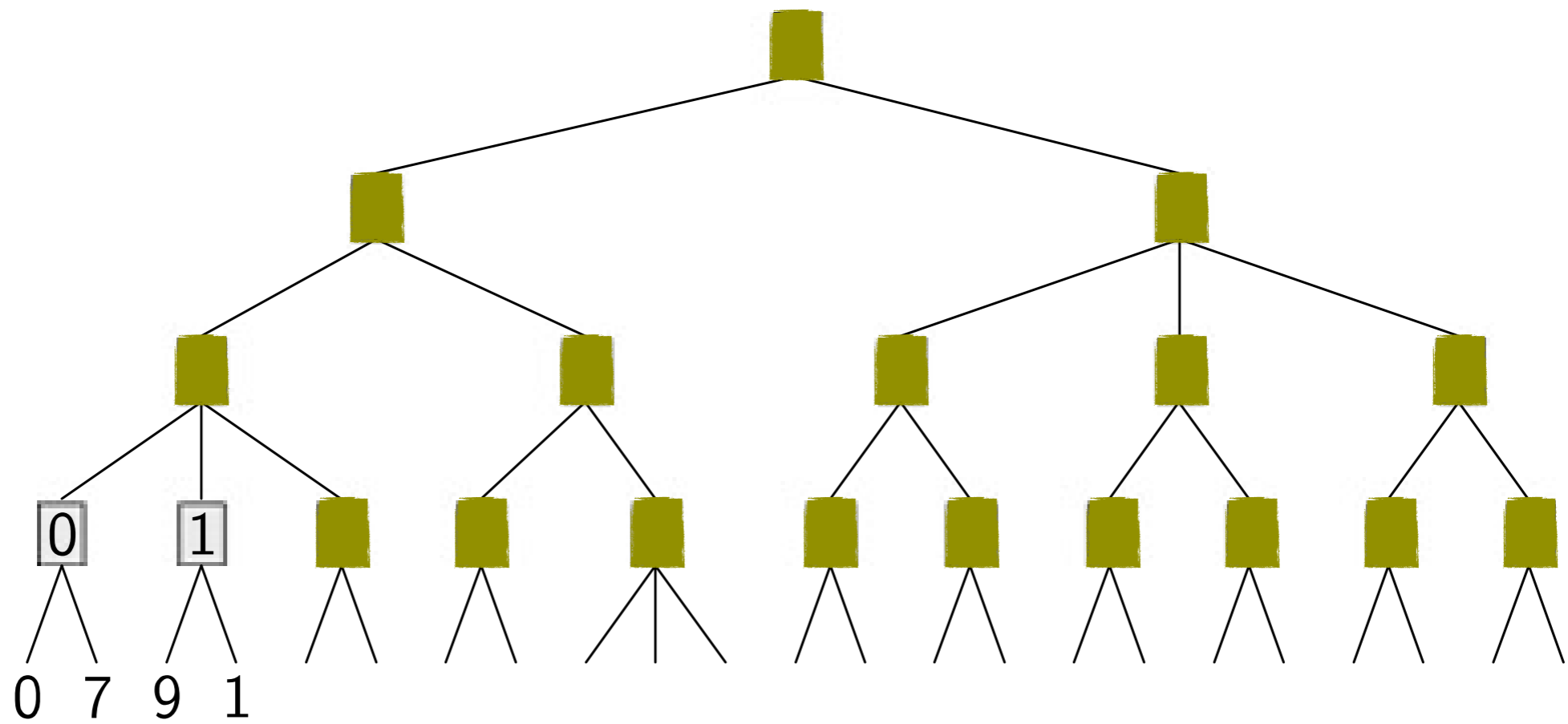
PELIPUU

MAX

MIN

MAX

MIN



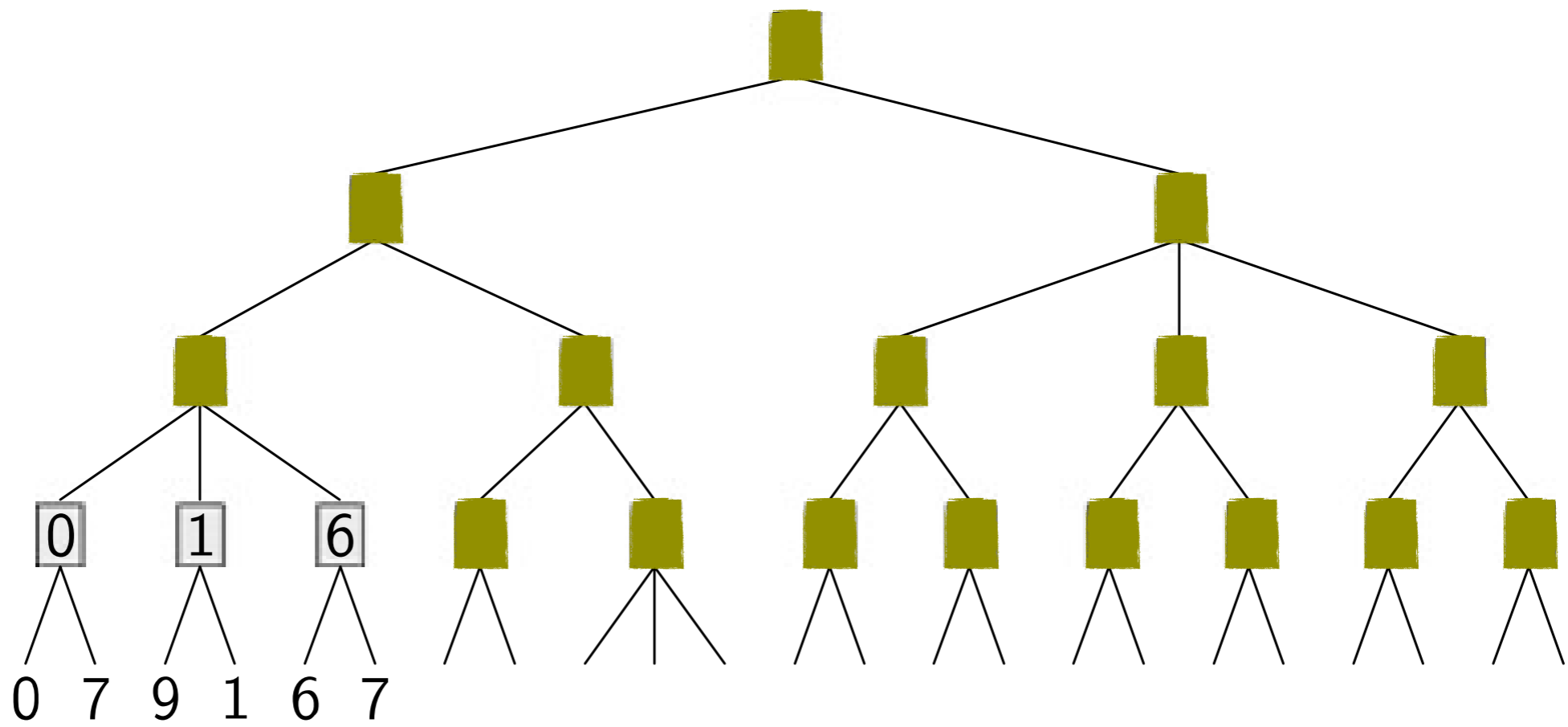
PELIPUU

MAX

MIN

MAX

MIN



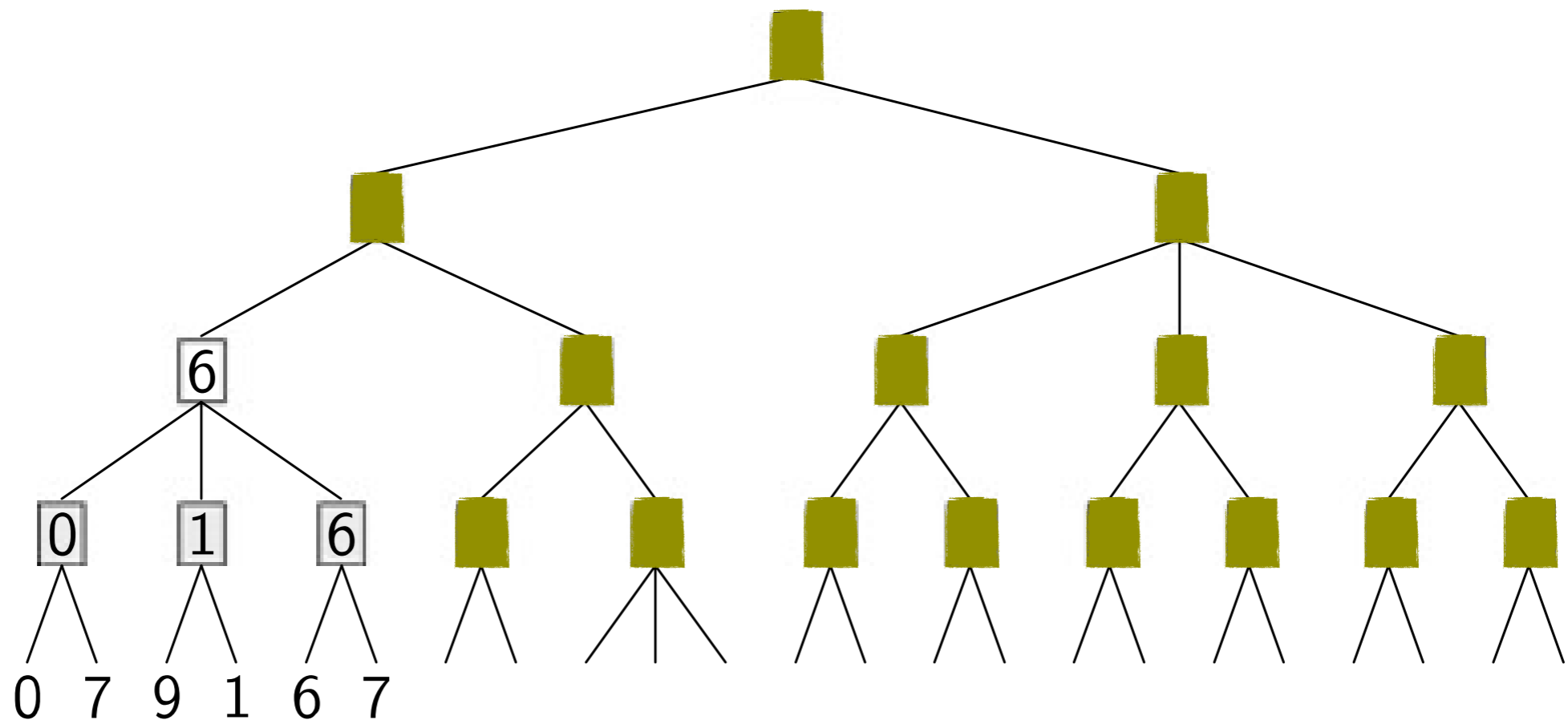
PELIPUU

MAX

MIN

MAX

MIN



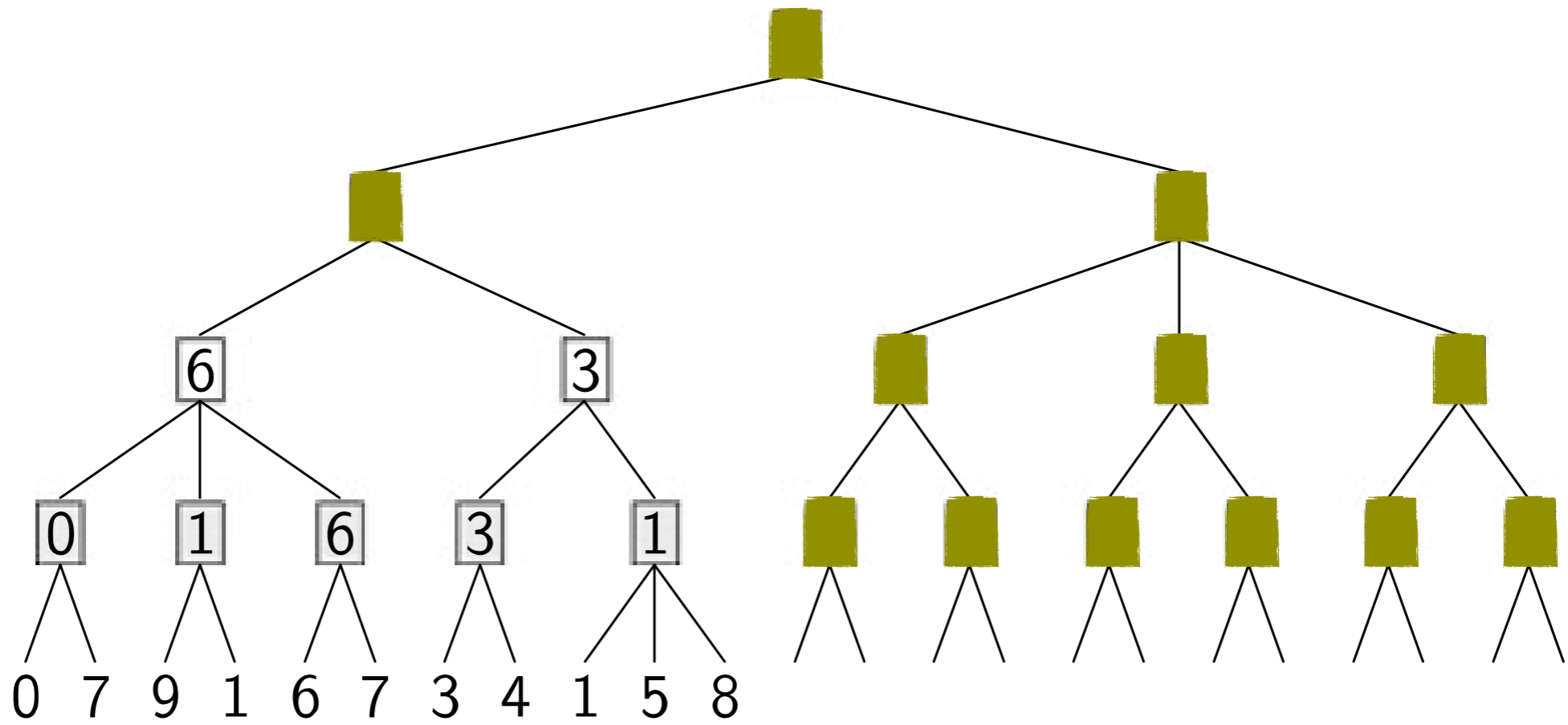
PELIPUU

MAX

MIN

MAX

MIN



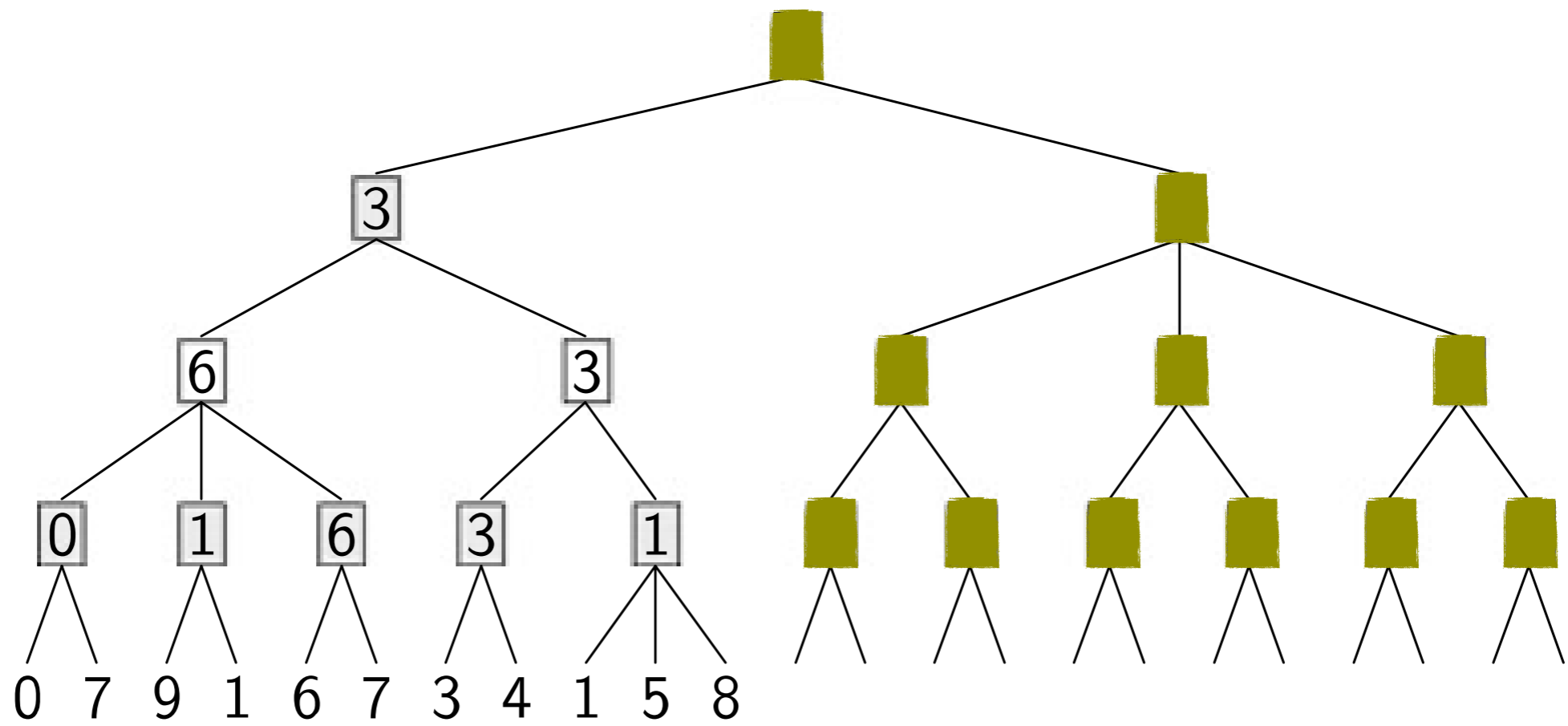
PELIPUU

MAX

MIN

MAX

MIN



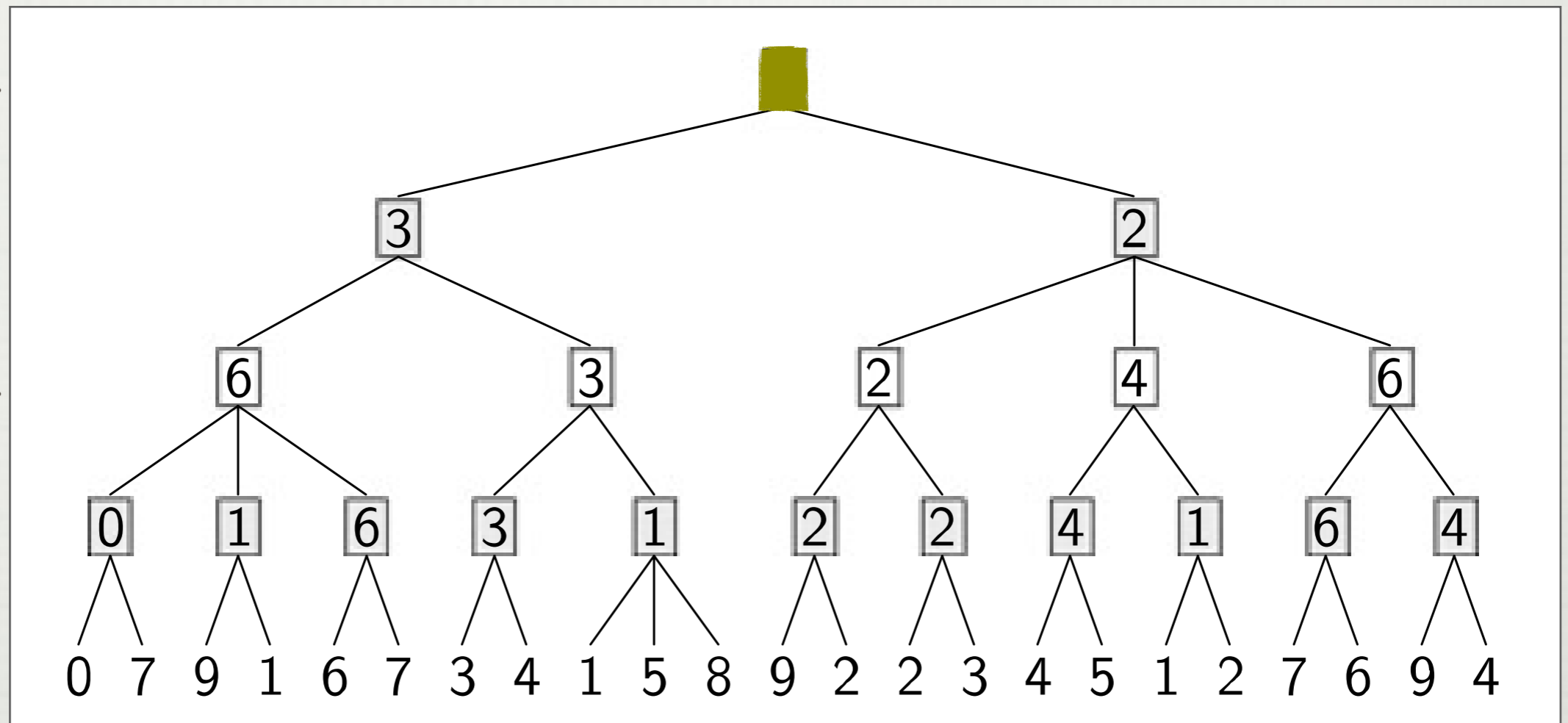
PELIPUU

MAX

MIN

MAX

MIN



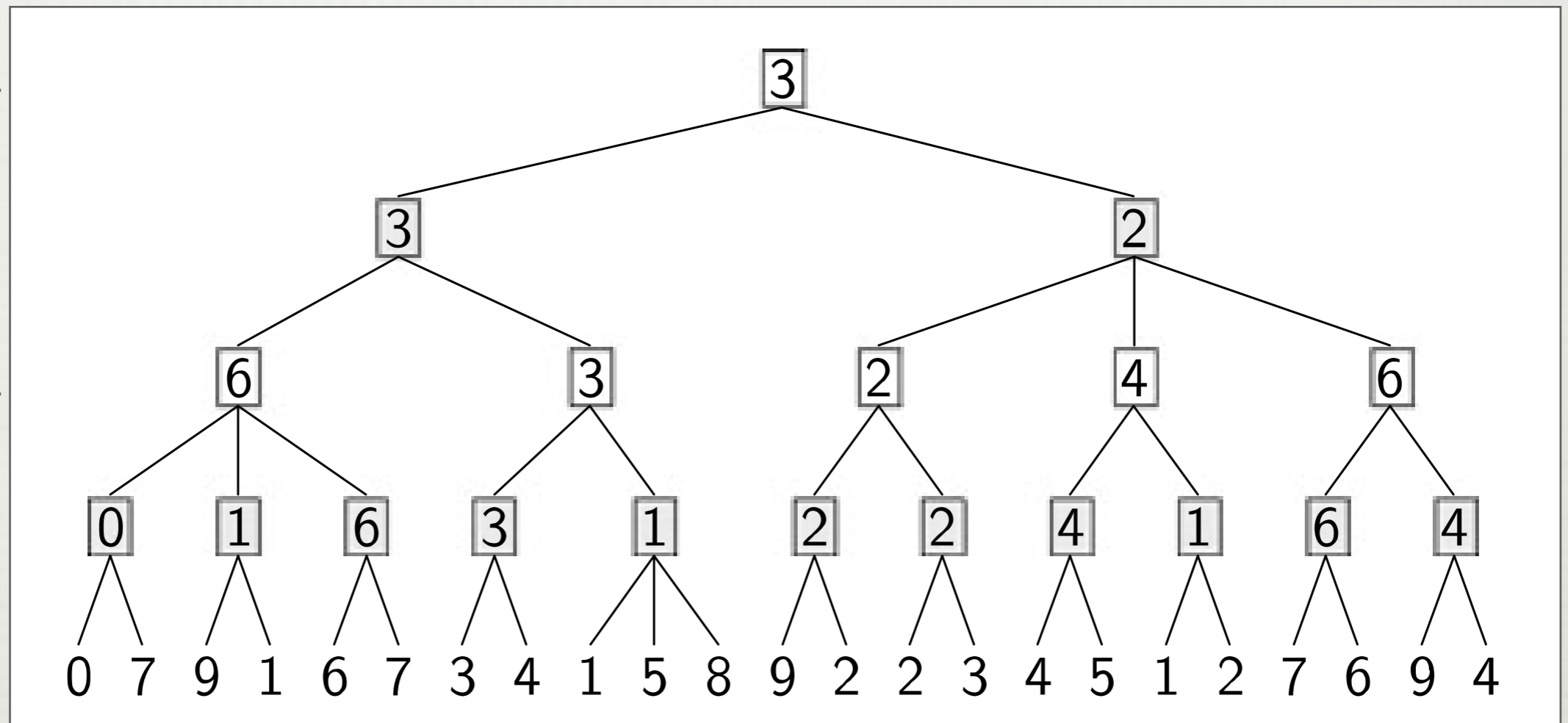
PELIPUU

MAX

MIN

MAX

MIN



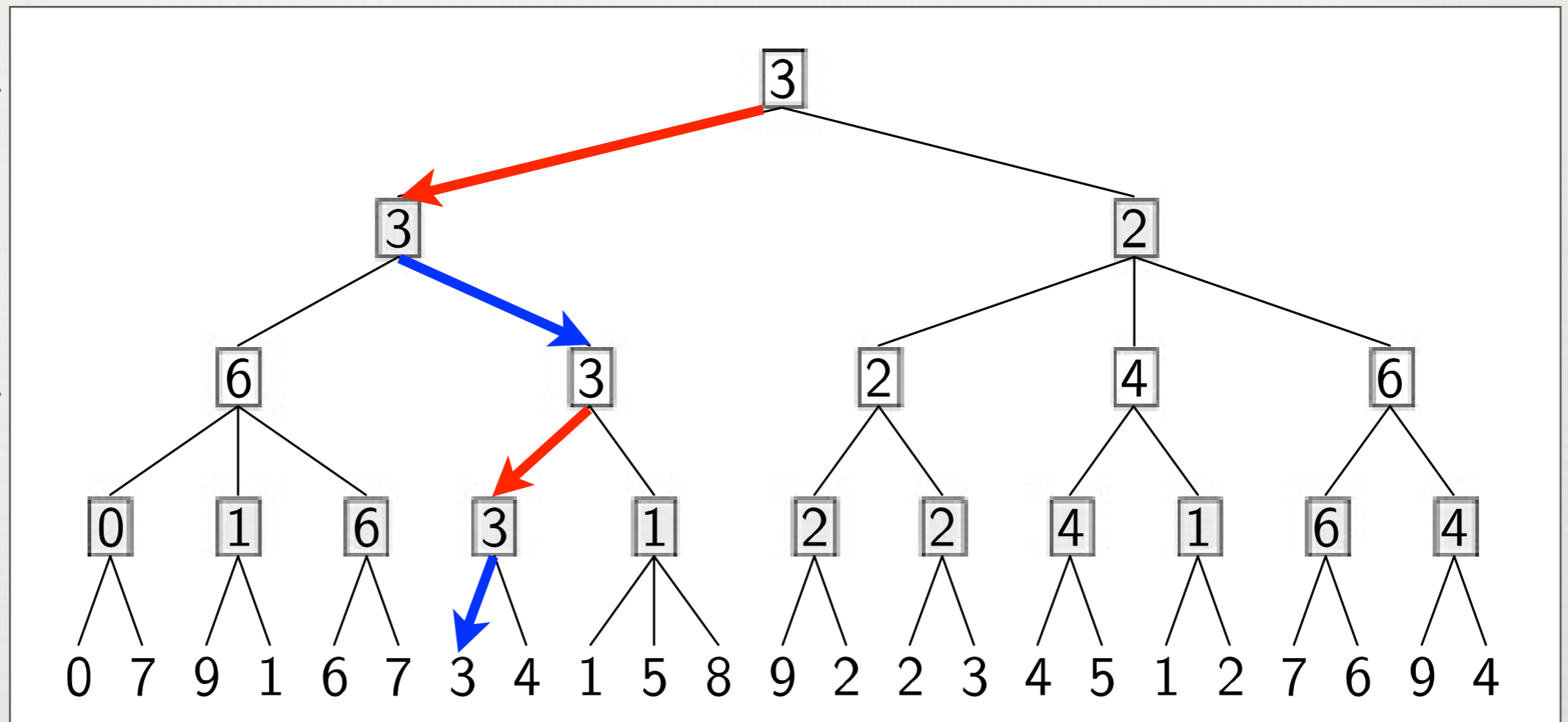
PELIPUU

MAX

MIN

MAX

MIN



MINIMAX

MAX-ARVO(Solmu)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = -\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MAX}(v, \text{MIN-ARVO}(\text{Lapsi}))$

return(v)

MIN-ARVO(Solmu)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MIN}(v, \text{MAX-ARVO}(\text{Lapsi}))$

return(v)

MINIMAX

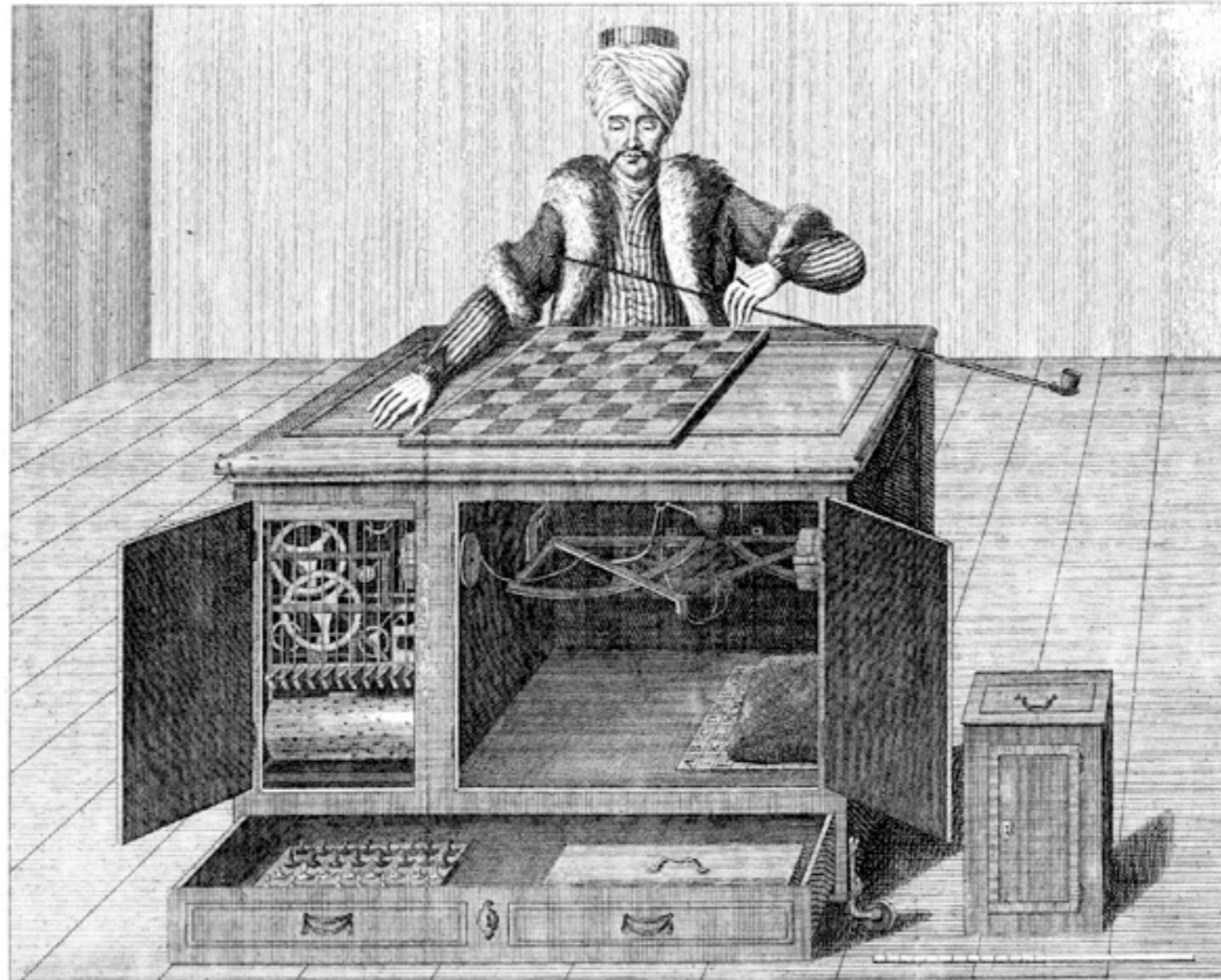
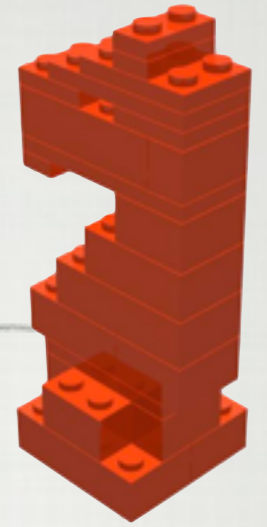
Game Demonstra

The purpose of this demonstration is to help you develop intuition for how minimax and alpha-beta search methods perform. The particular problem solved is that of finding the best move in a game.

The `search type` menu item on the menu bar enables you to see either the minimax method working alone or together with the alpha-beta method.

The `beta type` item

SHAKKI

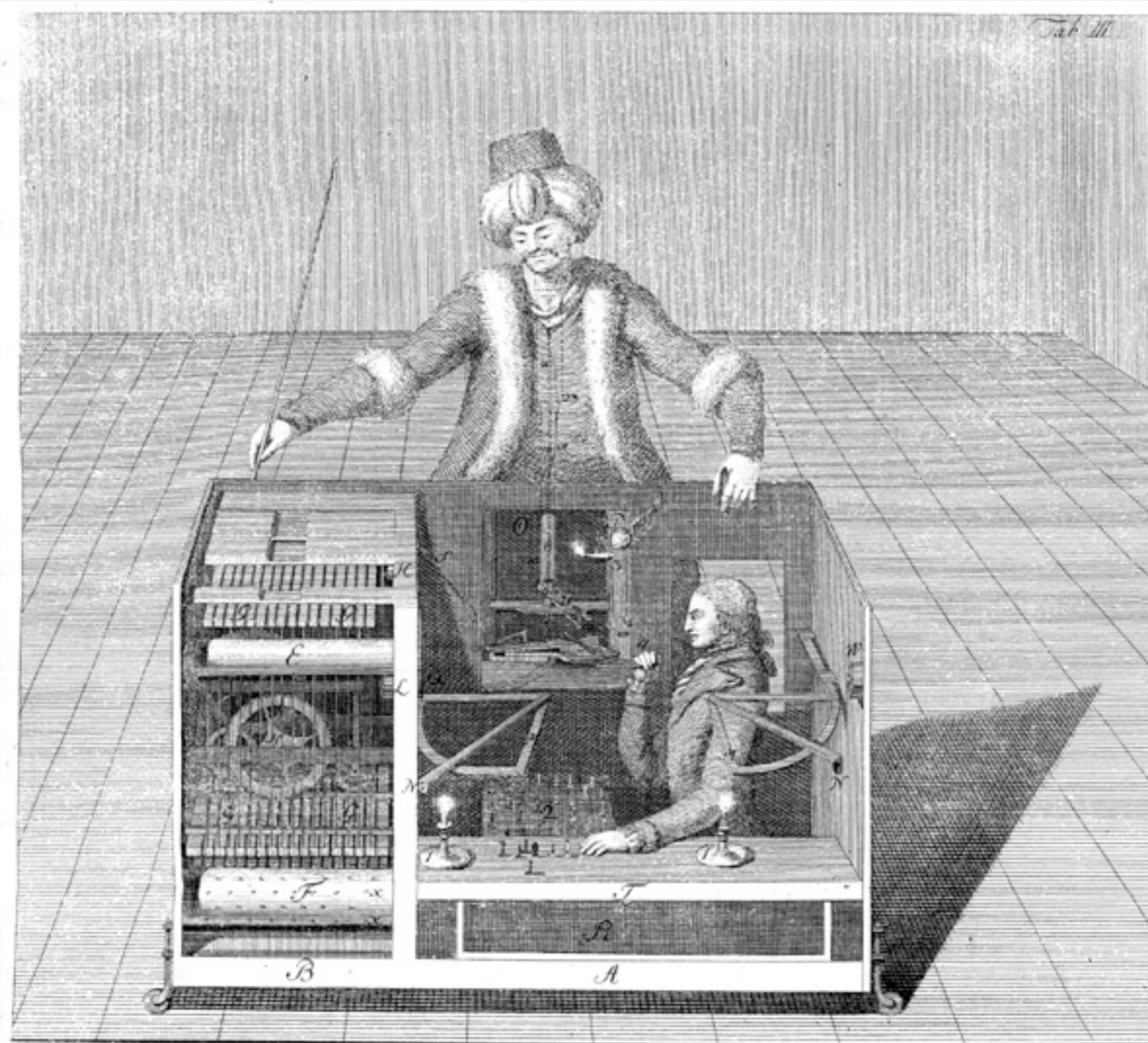
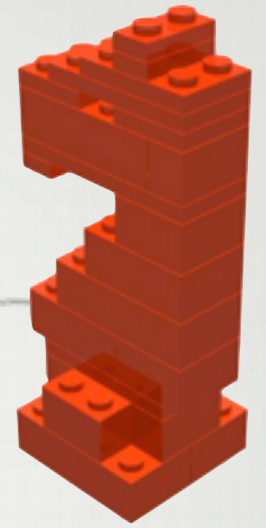


W. de Kempelen del. Che a Mechel, exaud. Basilea. P. G. Piatz, fecit.
Der Schach-Spieler, wie er vor dem Spiele gezeiget wird von vorn. Le Joueur d'Échecs, tel qu'on le montre avant le jeu, par devant.

SHAKKI

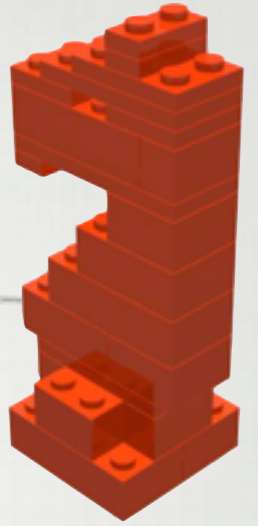
The image shows a screenshot of a YouTube video player. At the top left is the YouTube logo. To its right is a search bar with the text "Search" and "Browse | Upload" links. Below the search bar is the video title "Deep Blue beat G. Kasparov in 1997" in bold black text. Under the title, the channel name "Eustake" is displayed in blue, followed by "6 videos" with a dropdown arrow and a "Subscribe" button. The video player itself is mostly black, indicating it is loading, with a white loading spinner in the center. At the bottom of the player, there is a control bar with a play/pause button, a volume icon, a progress bar showing "0:00 / 6:06", a Creative Commons license icon, "360p" resolution, and other standard video controls. On the right side of the player, the word "Suggesti" is partially visible, indicating a suggestions sidebar.

SHAKKI



SHAKKI

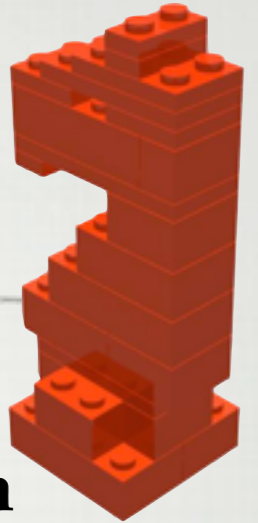
(NICE TO KNOW: EI TARVITSE OPETELLA)



- 1769** **Wolfgang von Kempelen** rakentaa "Turkin"
- 1912** **L. Torres y Quevedo** rakentaa koneen kuningas&torni vs kuningas -loppupeleihin
- 1948** **Norbert Wiener** esittää syvyysrajoitetun minimax-algoritmin heuristisella arviontifunktiolla
- 1950** **Claude Shannon** julkaisee artikkelin "Programming a Computer for Playing Chess"
- 1951** **Alan Turing** kehittää ensimmäisen algoritmin, joka pystyy pelaamaan kokonaisen shakkiottelun
- 1956** Los Alamos chess: ensimmäinen tietokoneohjelma, joka pelaa (yksinkertaistettua) shakkia
- 1956** **John McCarthy** keksii alpha-beta-karsinnan
- 1957** Ensimmäiset oikeaa shakkia pelaavat ohjelmat
- 1966-67** Ensimmäiset tietokoneohjelmien väliset ottelut (Moskova voittaa.)

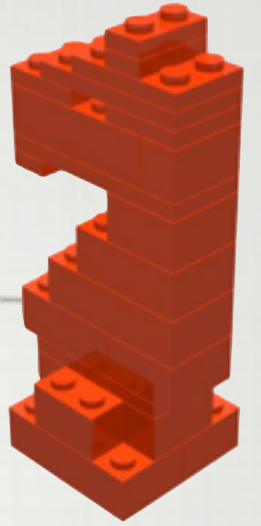
SHAKKI

(NICE TO KNOW: EI TARVITSE OPETELLA)



- 1967** Ensimmäinen tietokoneohjelman voitto turnauksessa.
- 1981** Cray Blitz voittaa Mississippin osavaltion mestaruuden ja saa ensimmäisenä tietokoneena mestarin statuksen.
- 1988** Deep Thought voittaa ensimmäistä kertaa suurmestarin turnauksessa.
- 1989** **Garry Kasparov** voittaa kaksi näytösottelua Deep Thoughtia vastaan.
- 1996** **Garry Kasparov** voittaa Deep Bluen kuuden pelin ottelussa.
- 1997** Deep Blue voittaa **Garry Kasparovin** kuuden pelin ottelussa.
- 2006** Deep Fritz voittaa maailmanmestari **Vladimir Kramnikin**.

SHAKKI



- * TILA: (LAUDAN TILANNE)
- * SIIRTYMÄT: (SALLITUT SIIRROT)
- * MENETELMÄ: SYVYYSSRAJOITETTU ALPHA-BETA-KARSINTA

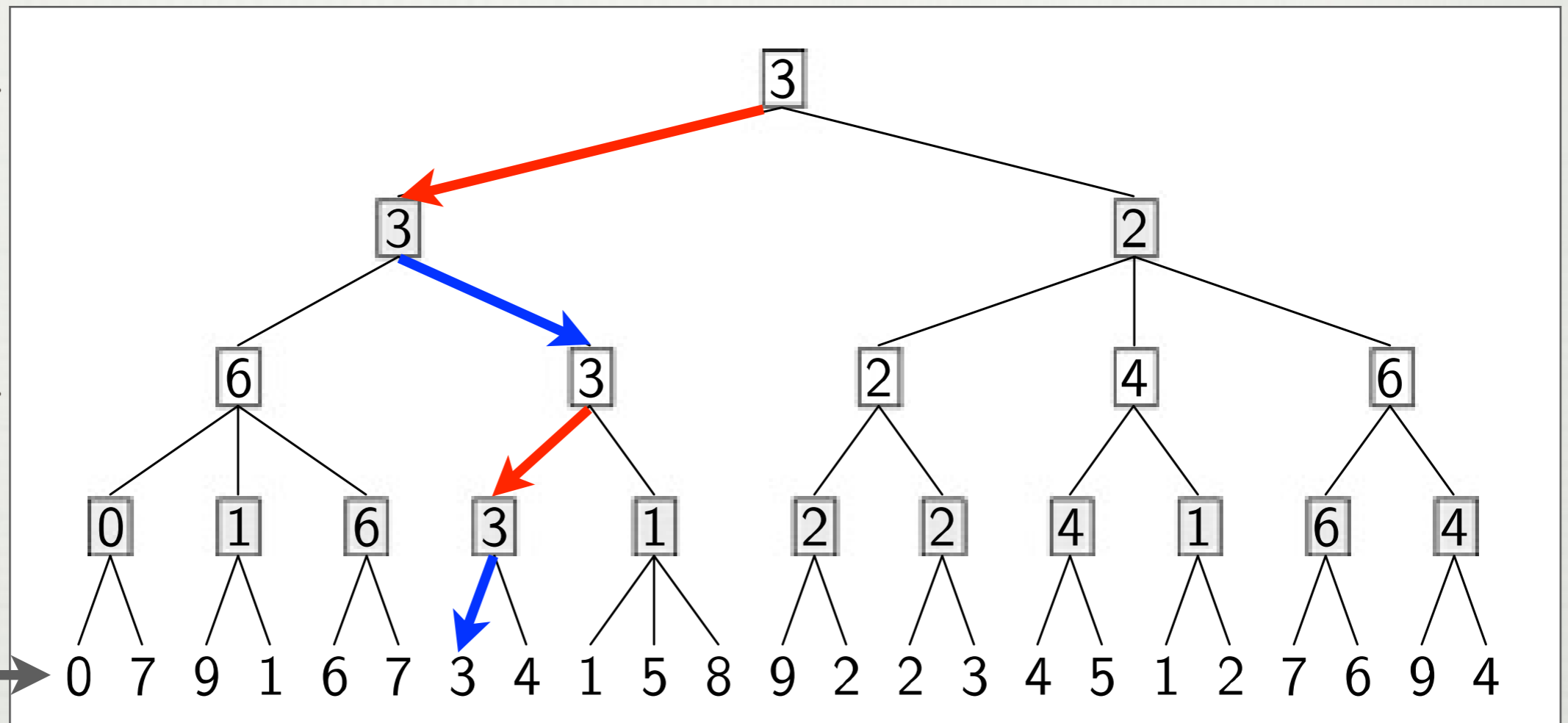
PELIPUU

MAX

MIN

MAX

MIN



ARVIOITA TILANTEEN HYVYYDESTÄ

SHAKKI

- * TILA: (LAUDAN TILANNE)
- * SIIRTYMÄT: (SALLITUT SIIRROT)
- * MENETELMÄ: SYVYYSRAJOITETTU ALPHA-BETA-KARSINTA
- * TEHTÄVÄ: SUUNNITTELE HEURISTINEN ARVIOINTIFUNKTIO

HEURISTIikkojen Valinnasta

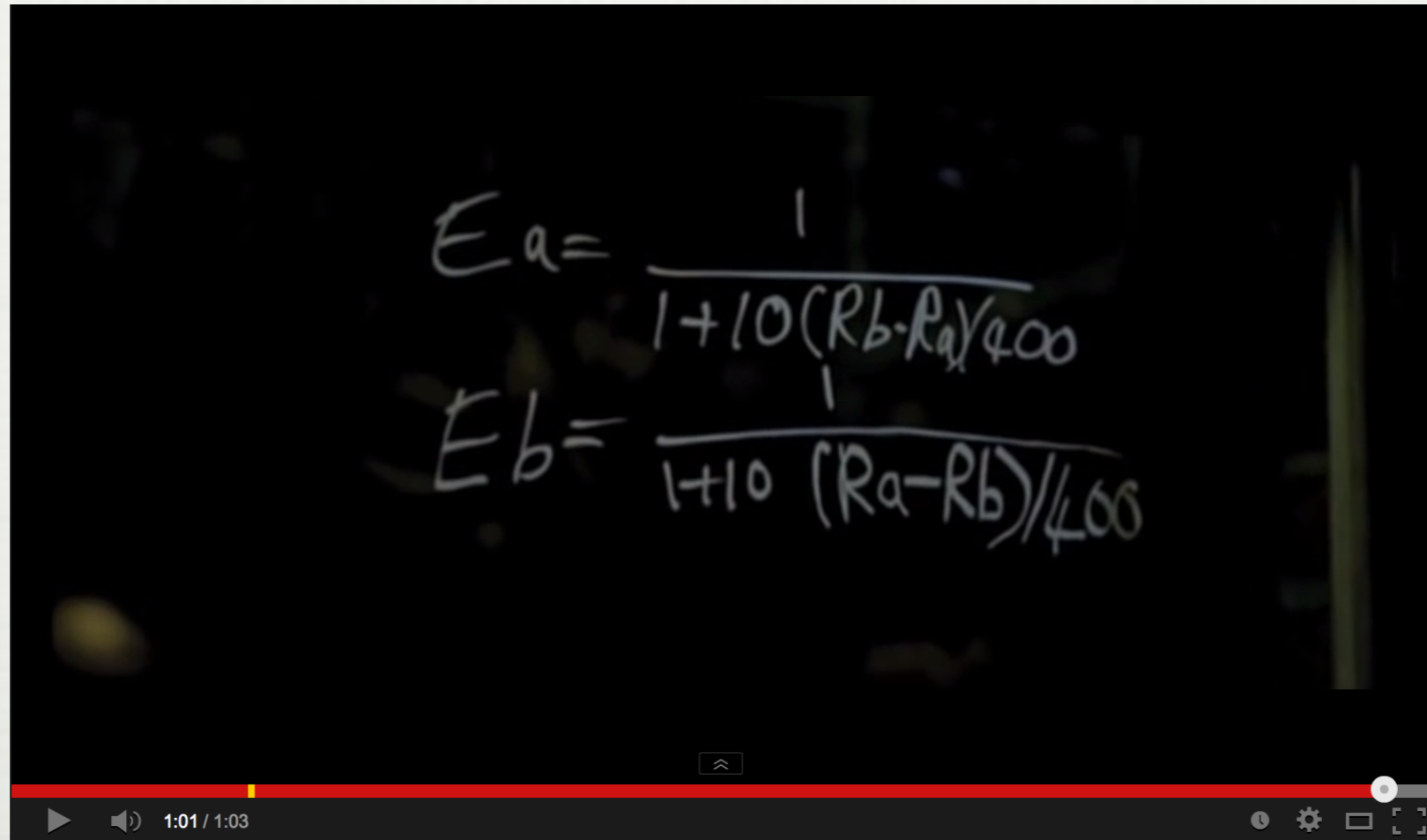
(NICE TO KNOW: EI TARVITSE OPETELLA)

- * HEURISTIikan Hyvyys vaikuttaa pelin tulokseen:
HYVÄ HEURISTIikka -> HYVÄ TULOS
- * VASTAAVASTI HEURISTIikan Hyvyyttä voi mitata tarkkailemalla pelien tuloksia:
HYVÄ TULOS -> HYVÄ HEURISTIikka
- * JOSKUS HYVÄKIN PELAAJA VOI SILTI HÄVITÄ HUONOMMALLEEN JA TOISINPÄIN, JOTEN ARVIOINTI EI OLE HELPPOA
- * YLEINEN MENETELMÄ HYVYYDEN ARVIOINTIIN:
ELO-RATING

HEURISTIIKKOJEN VALINNASTA

(NICE TO KNOW: EI TARVITSE OPETELLA)

ELO RATING - "I NEED THE ALGORITHM"



The image shows a blackboard with two ELO rating formulas written in white chalk. The first formula is $E_a = \frac{1}{1 + 10(R_b - R_a)/400}$ and the second is $E_b = \frac{1}{1 + 10(R_a - R_b)/400}$. Below the formulas is a video player interface with a red progress bar, a play button, a volume icon, and a timestamp of 1:01 / 1:03.

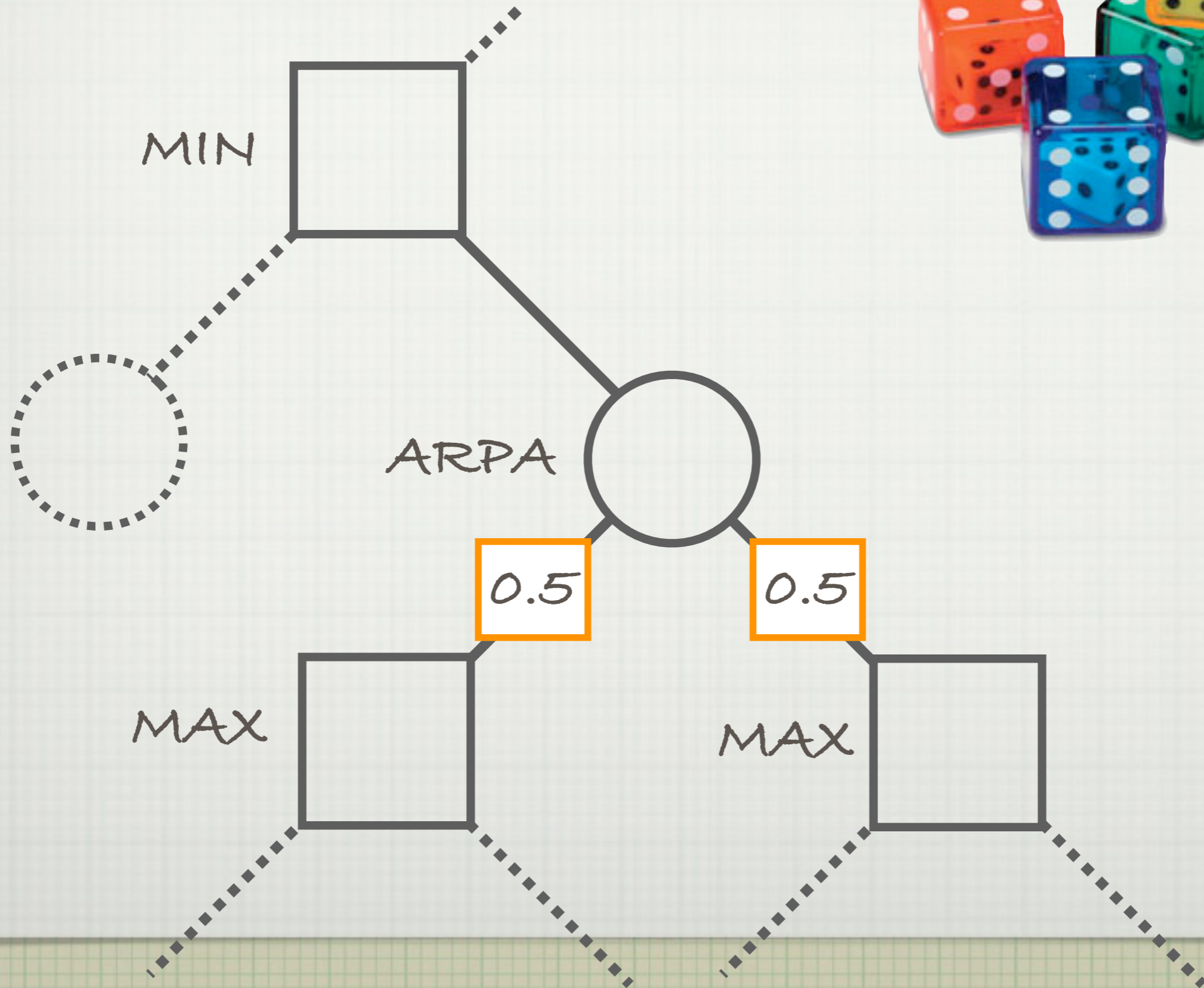
$$E_a = \frac{1}{1 + 10(R_b - R_a)/400}$$
$$E_b = \frac{1}{1 + 10(R_a - R_b)/400}$$

The Social
Network



© Columbia Pictures, Inc.

ARPAPELIT



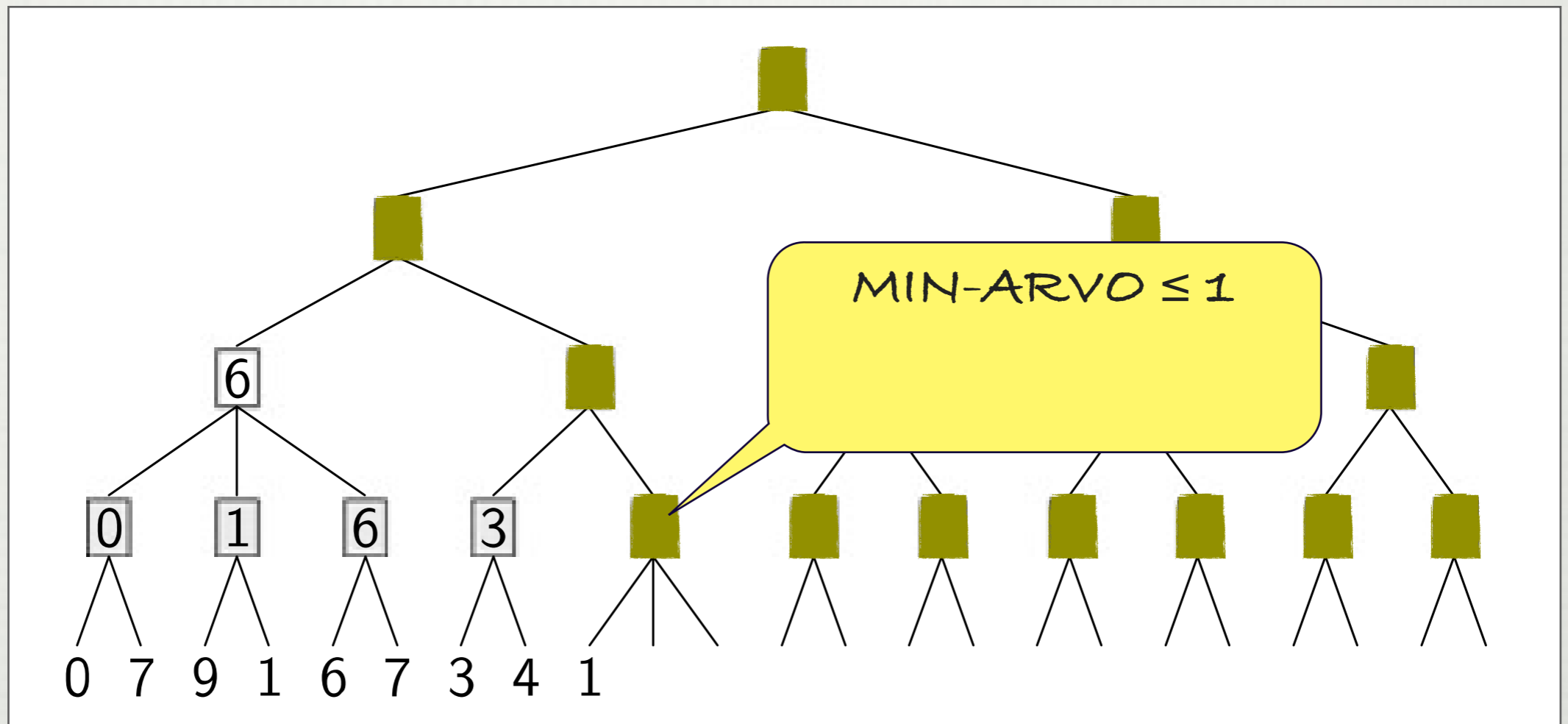
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



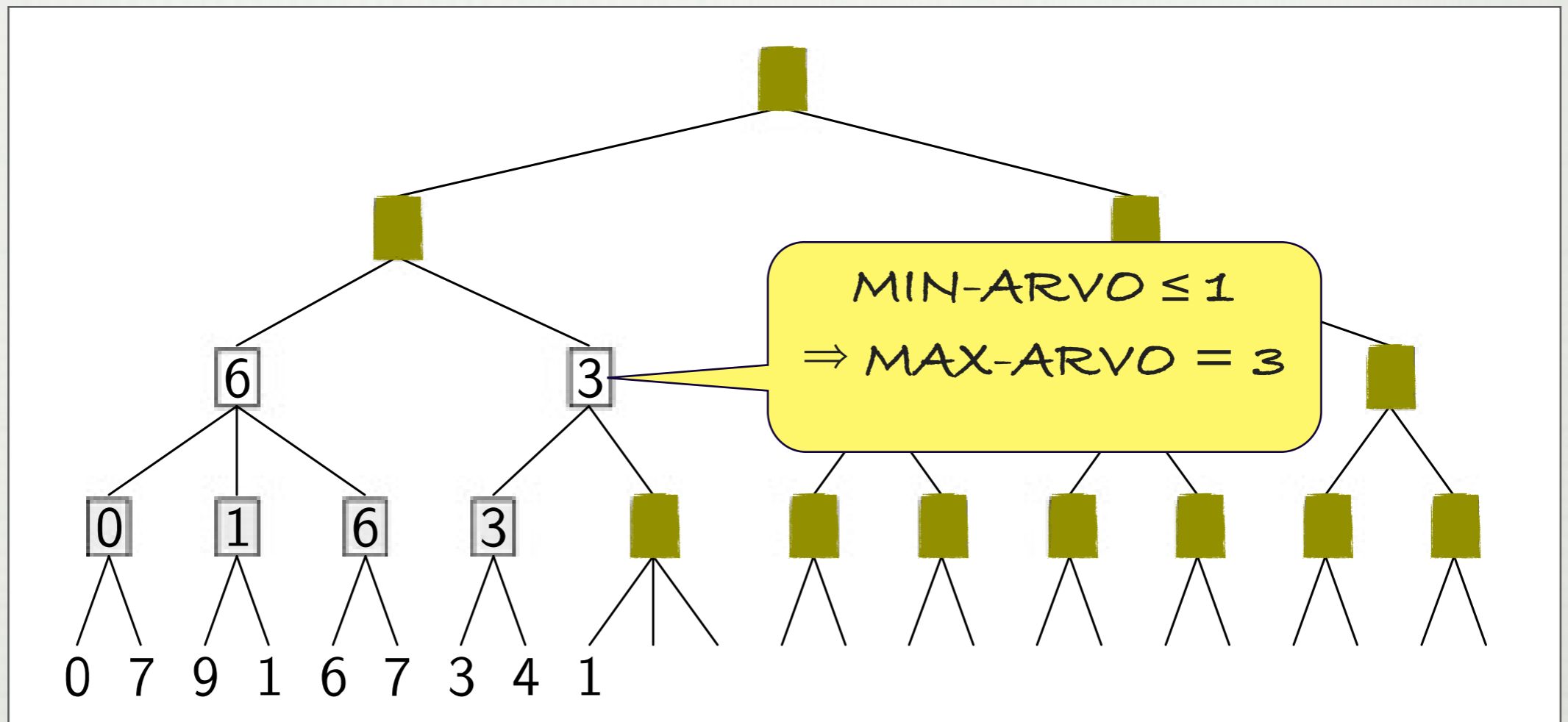
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



ALPHA-BETA-KARSINTA

MAX-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = -\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MAX}(v, \text{MIN-ARVO}(\text{Lapsi}, \alpha, \beta))$

if $v \geq \beta$ **return**(v)

$\alpha = \text{MAX}(\alpha, v)$

return(v)

MIN-PELAAJAN
TOISTAISEKSI
PARAS ARVO

MAX-PELAAJAN
TOISTAISEKSI
PARAS ARVO

ALPHA-BETA-KARSINTA

MAX-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = -\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MAX}(v, \text{MIN-ARVO}(\text{Lapsi}, \alpha, \beta))$

if $v \geq \beta$ **return**(v)

$\alpha = \text{MAX}(\alpha, v)$

return(v)

MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MIN}(v, \text{MAX-ARVO}(\text{Lapsi}, \alpha, \beta))$

if $v \leq \alpha$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

return(v)

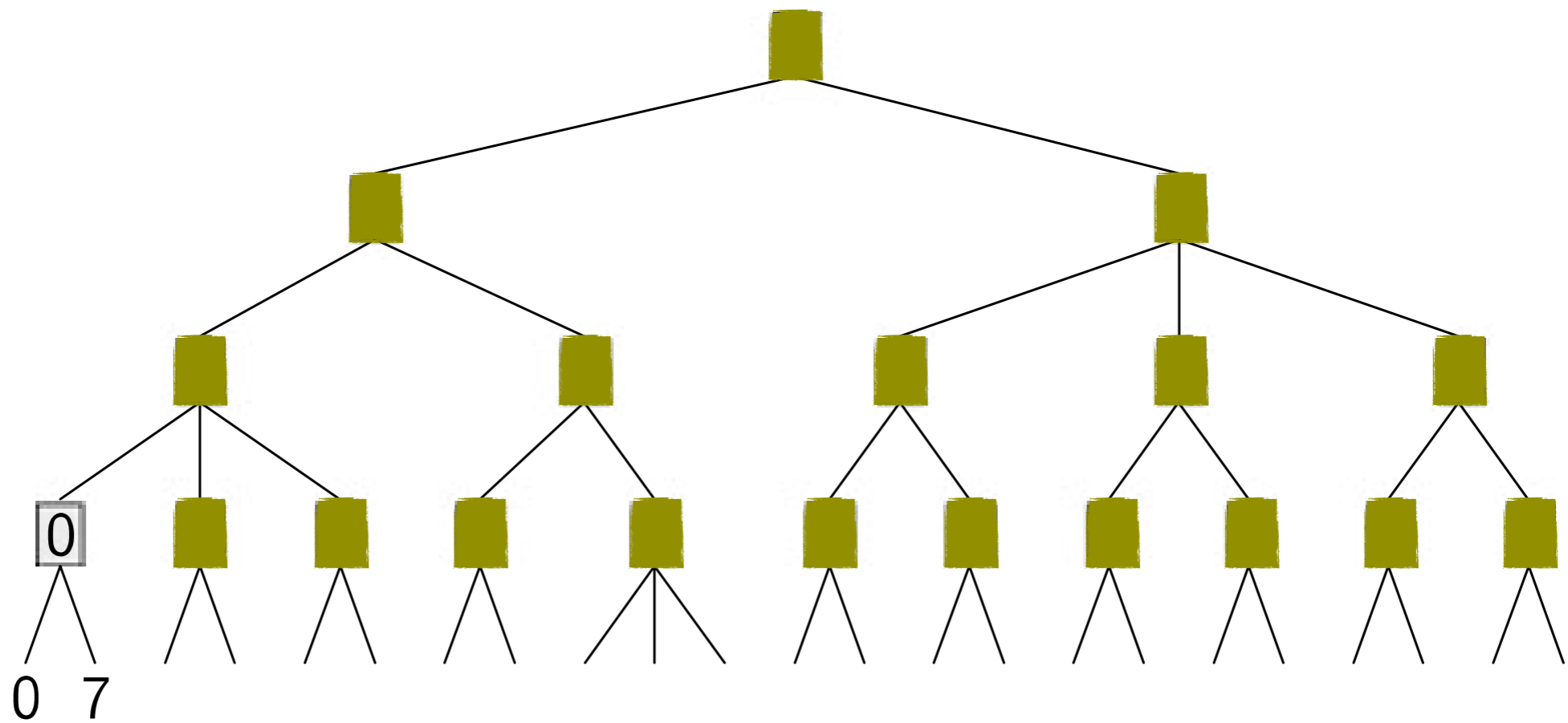
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



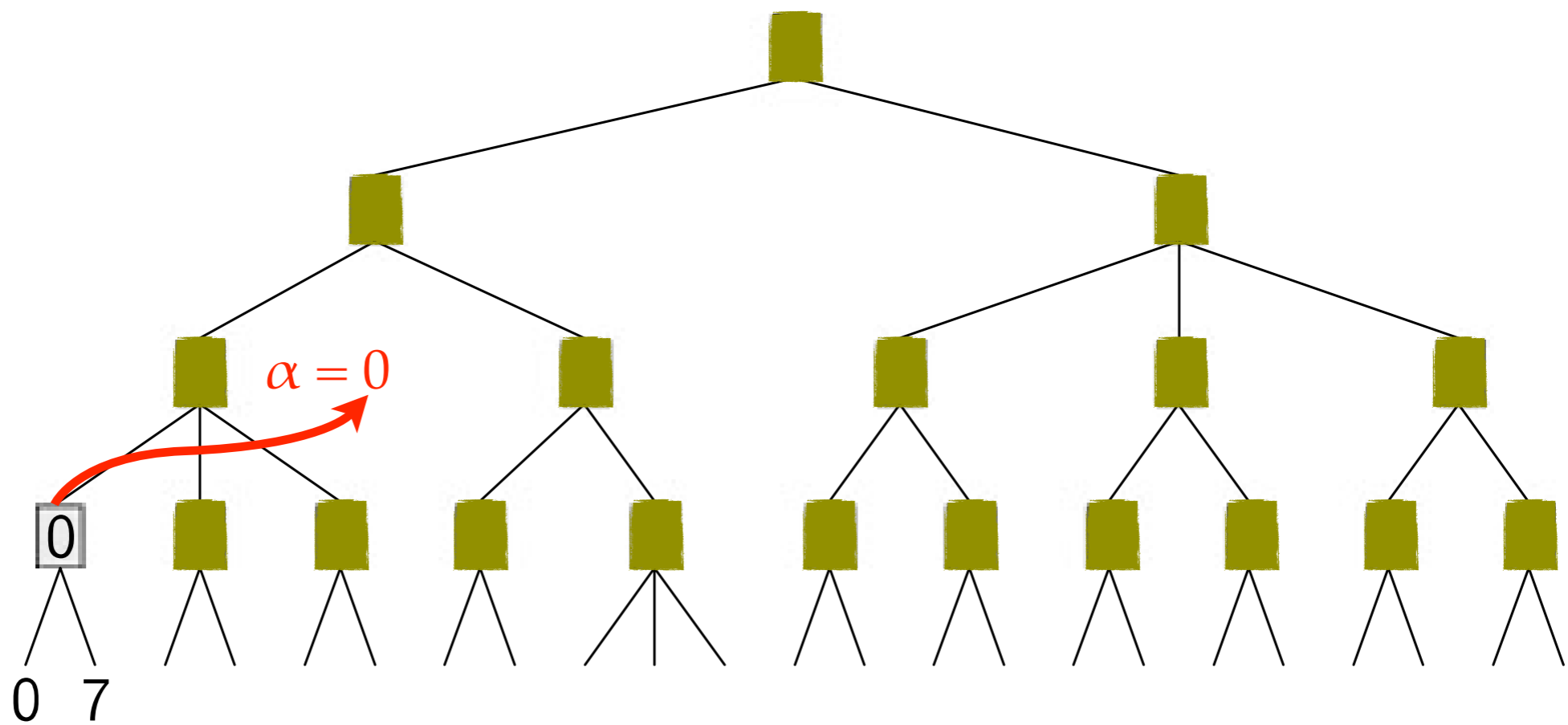
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



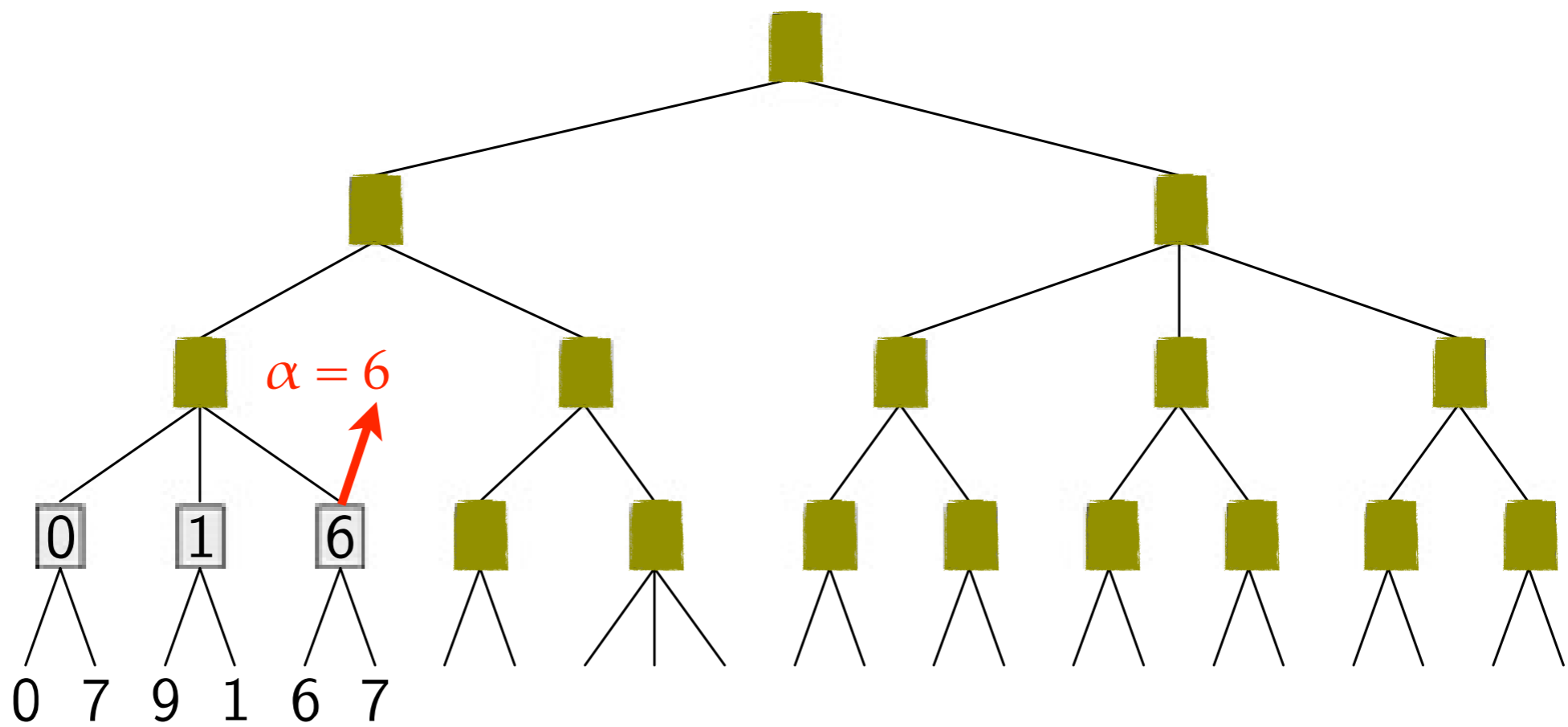
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



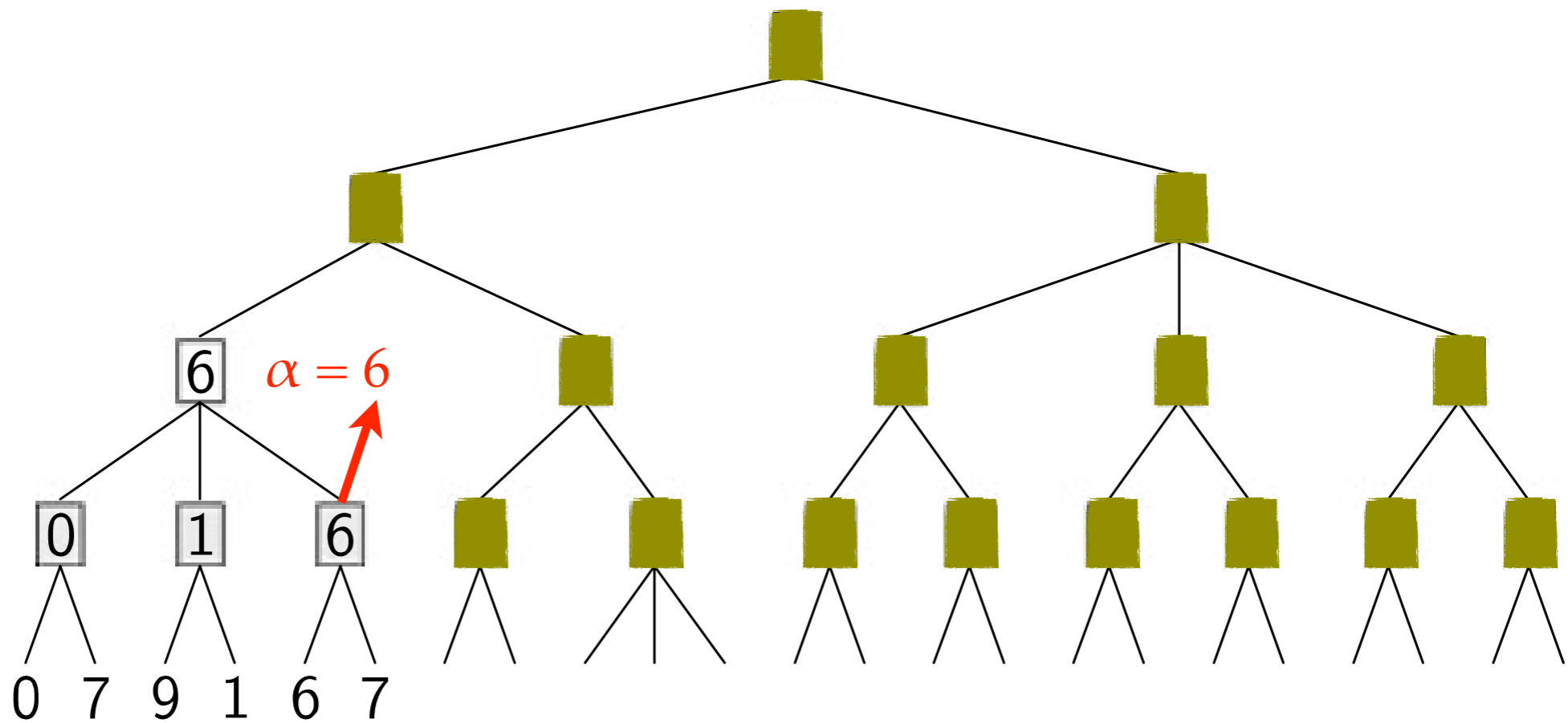
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



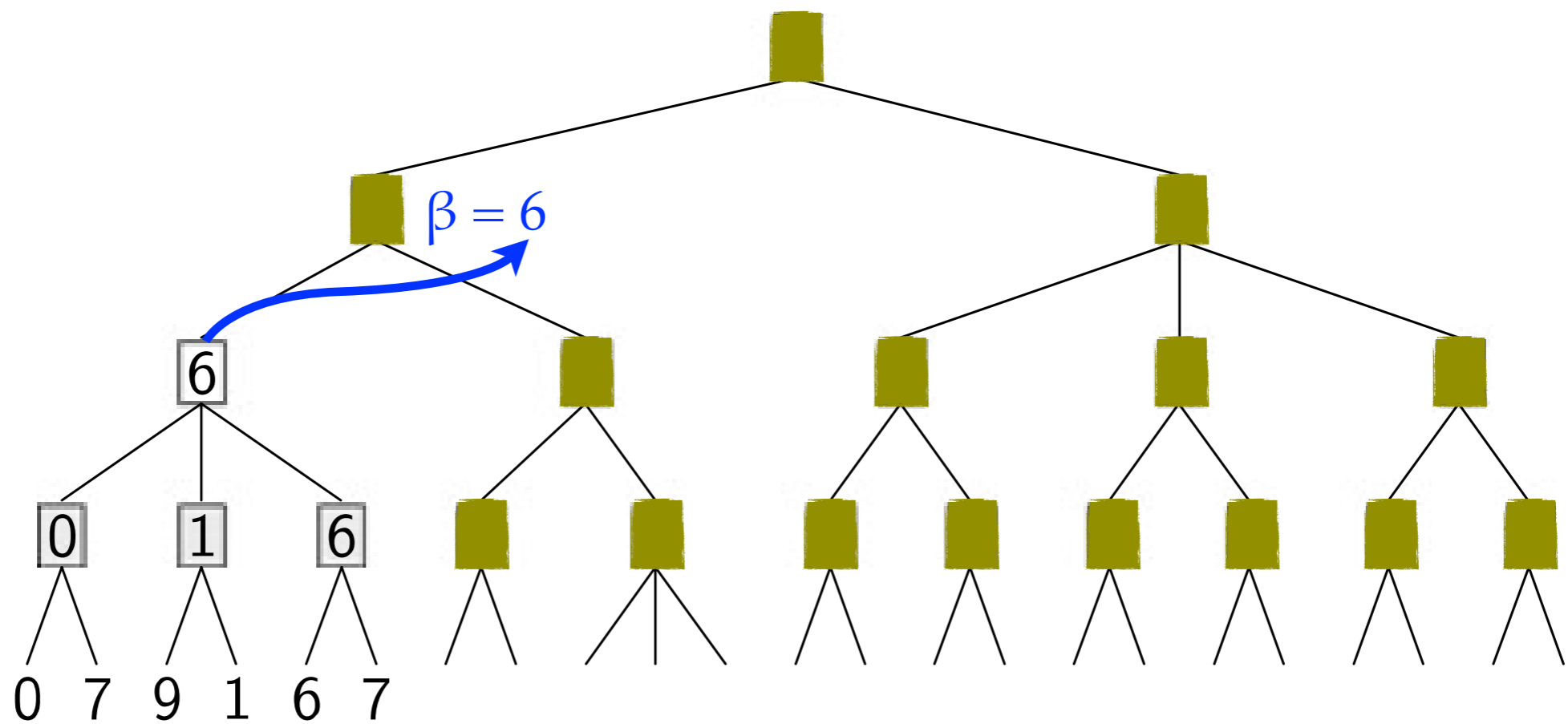
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



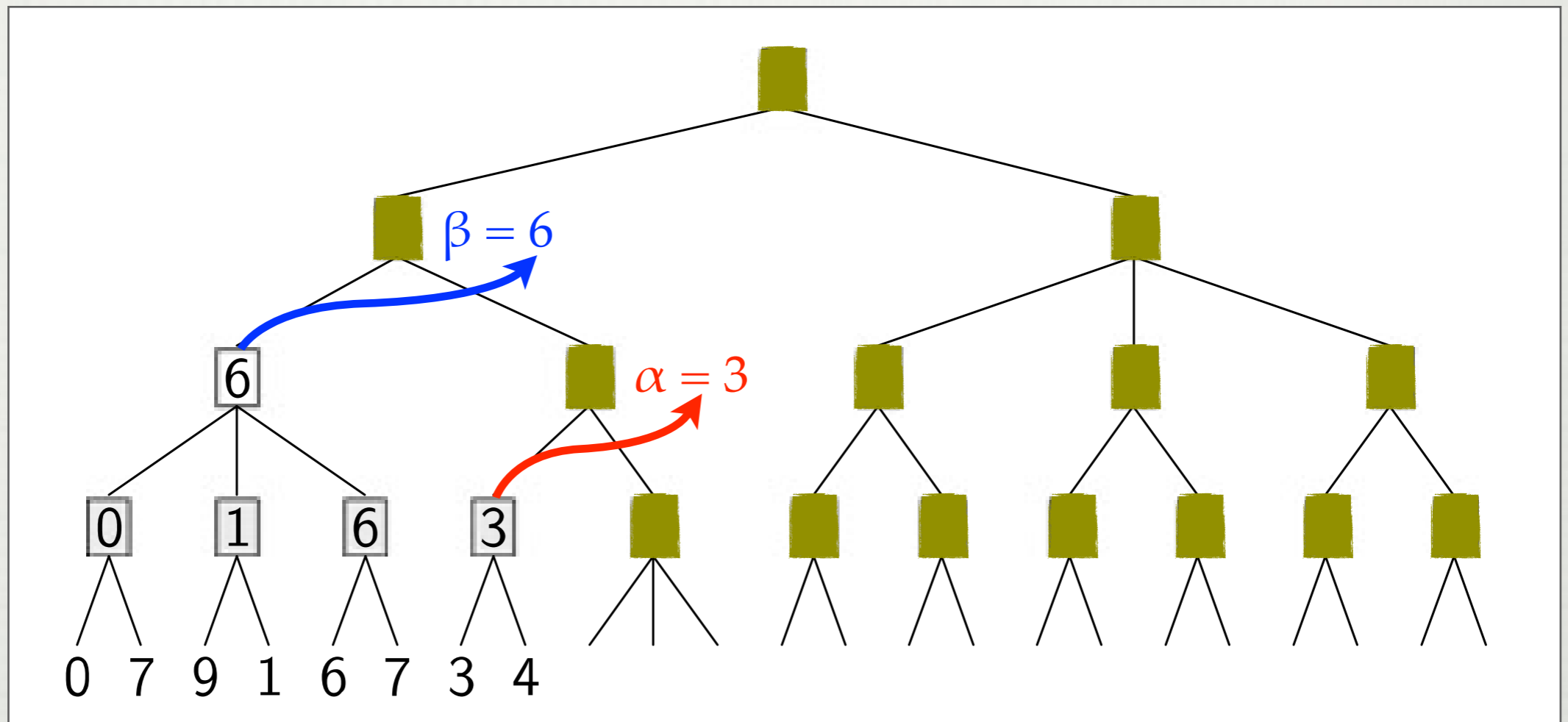
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



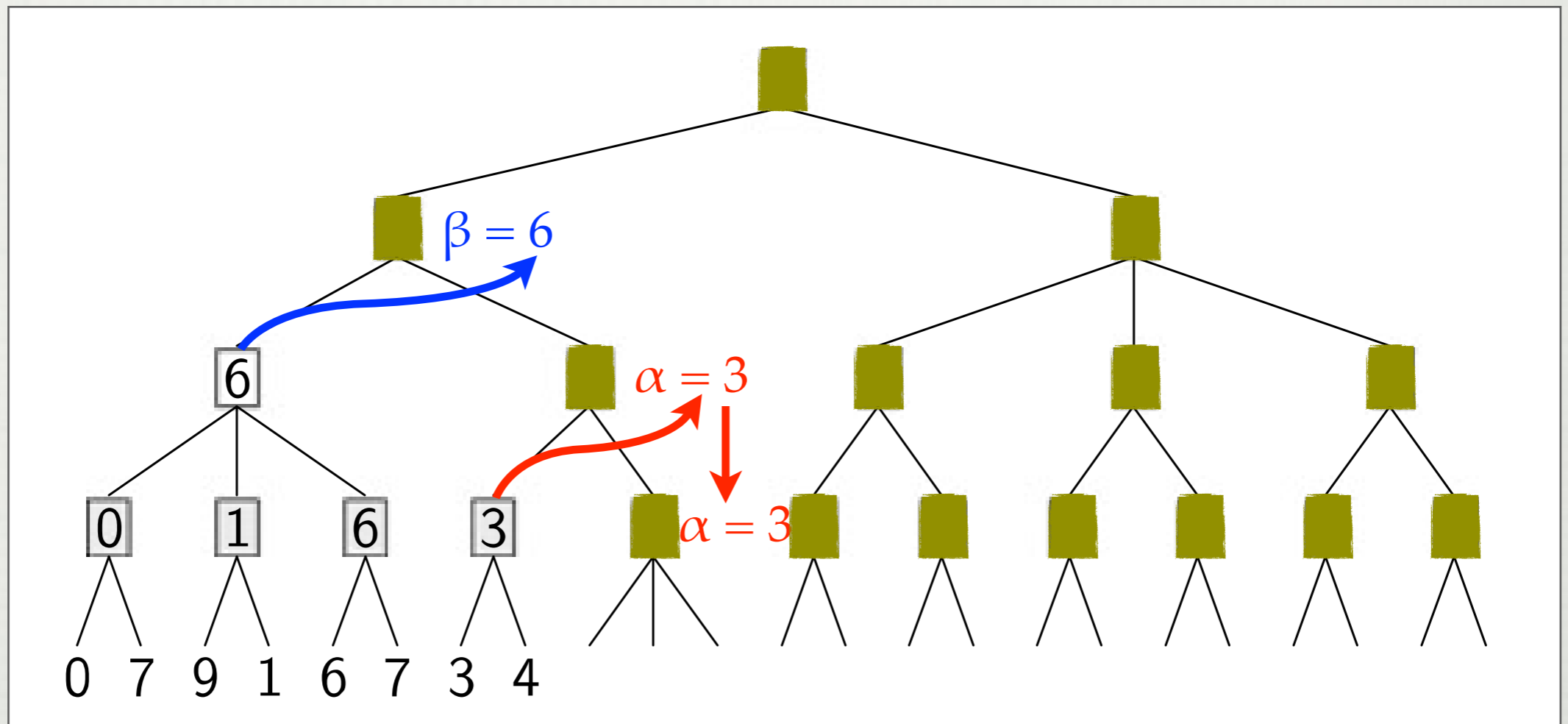
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



ALPHA

MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MIN}(v, \text{MAX-ARVO}(\text{Lapsi}, \alpha, \beta))$

if $v \leq \alpha$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

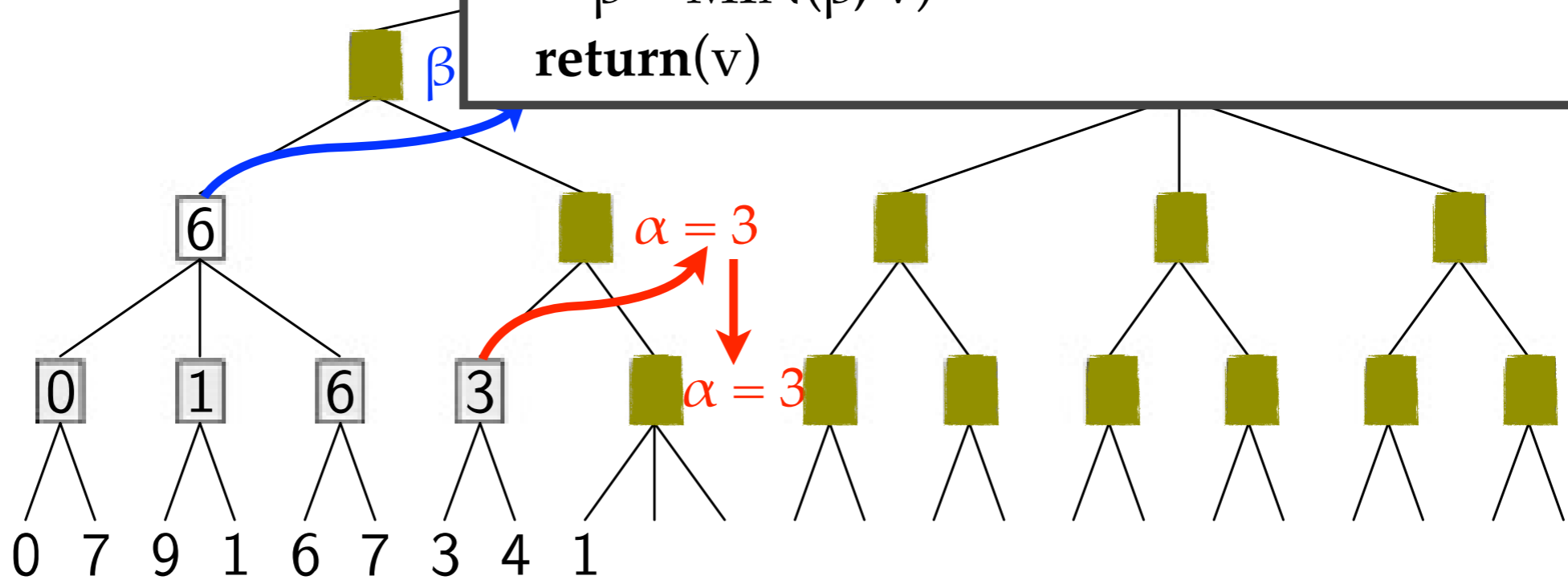
return(v)

MAX

MIN

MAX

MIN



ALPHA

MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 1$

if $v \leq \alpha$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

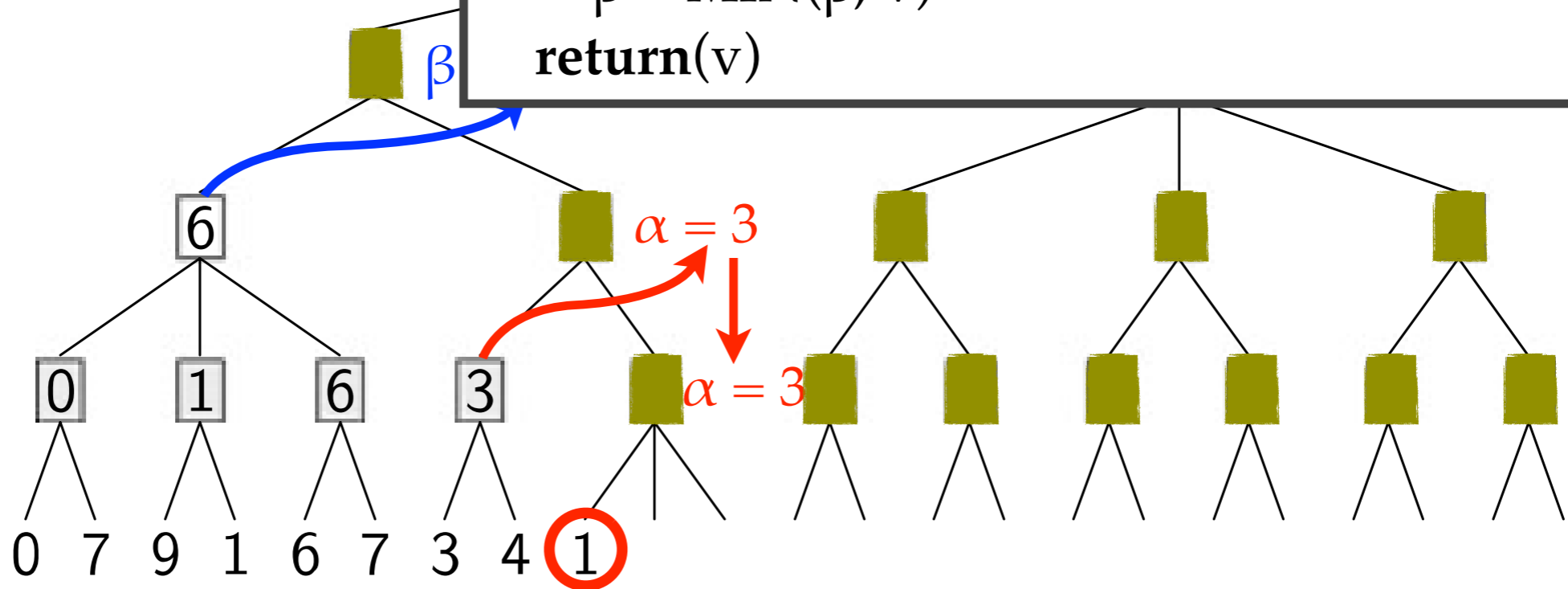
return(v)

MAX

MIN

MAX

MIN



ALPHA

MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 1$

if $1 \leq 3$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

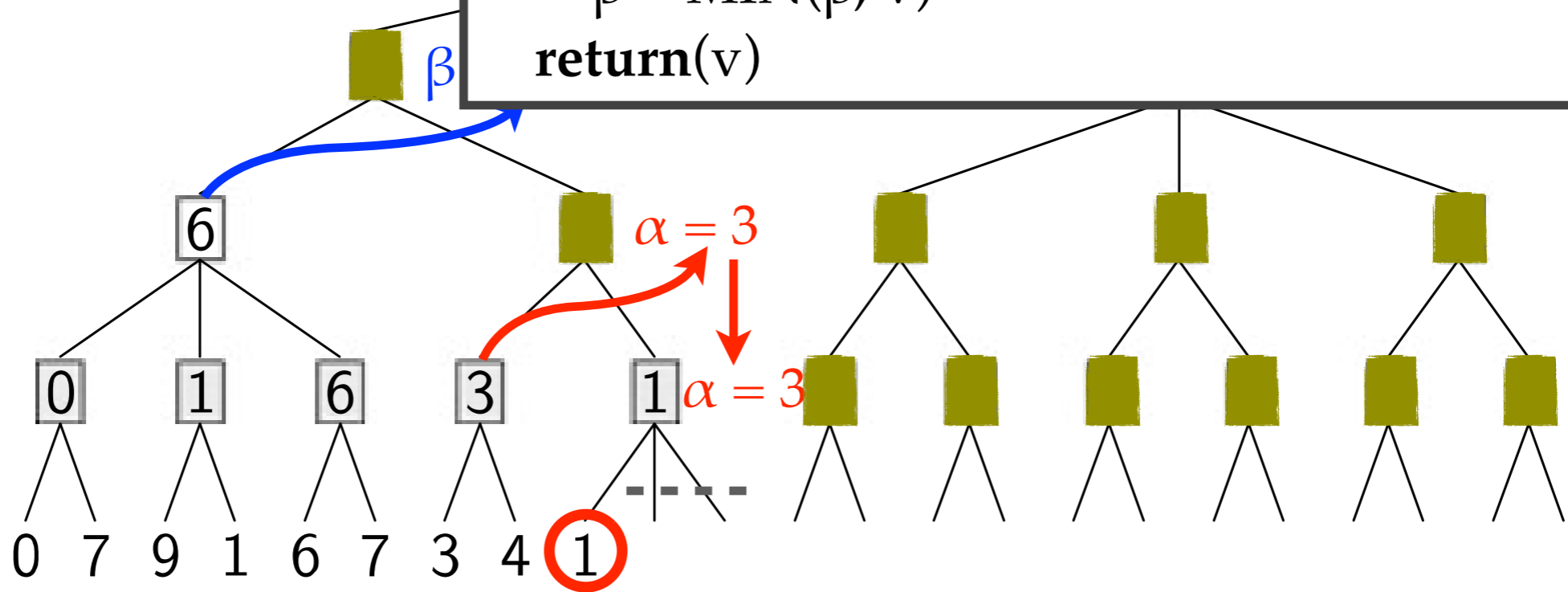
return(v)

MAX

MIN

MAX

MIN



ALPHA

MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 1$

if $1 \leq 3$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

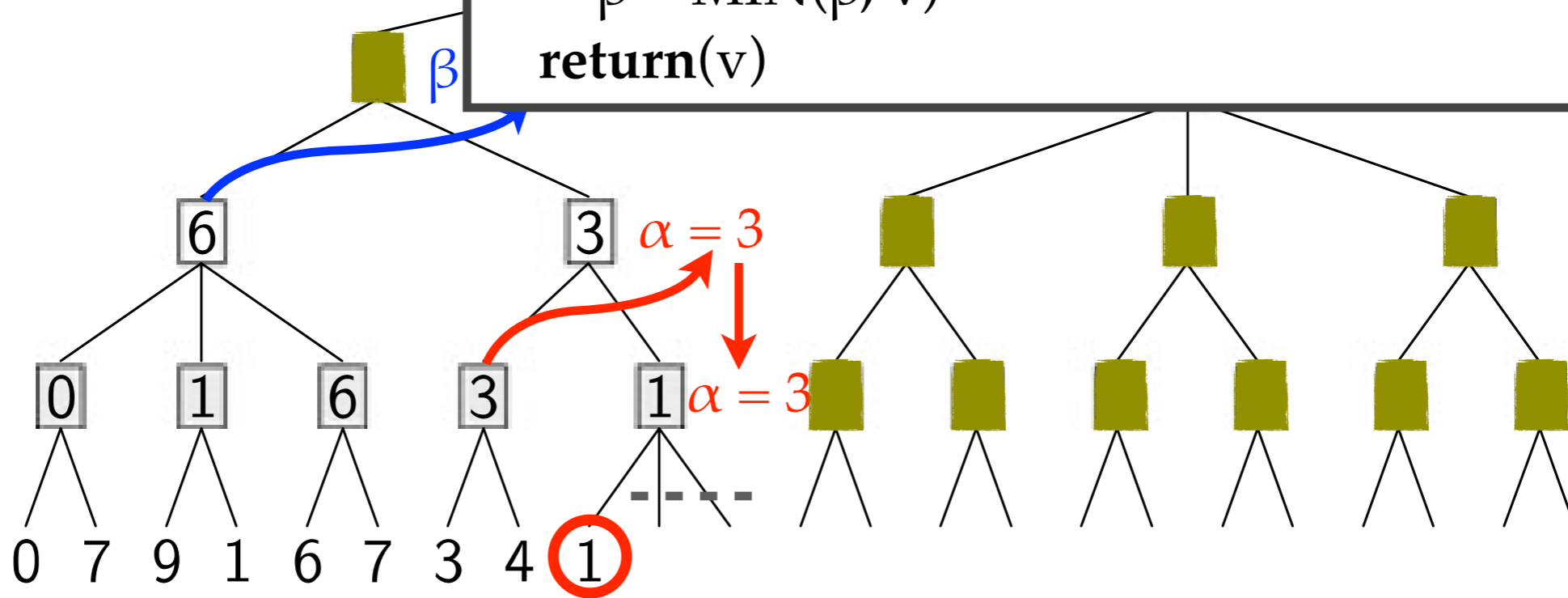
return(v)

MAX

MIN

MAX

MIN



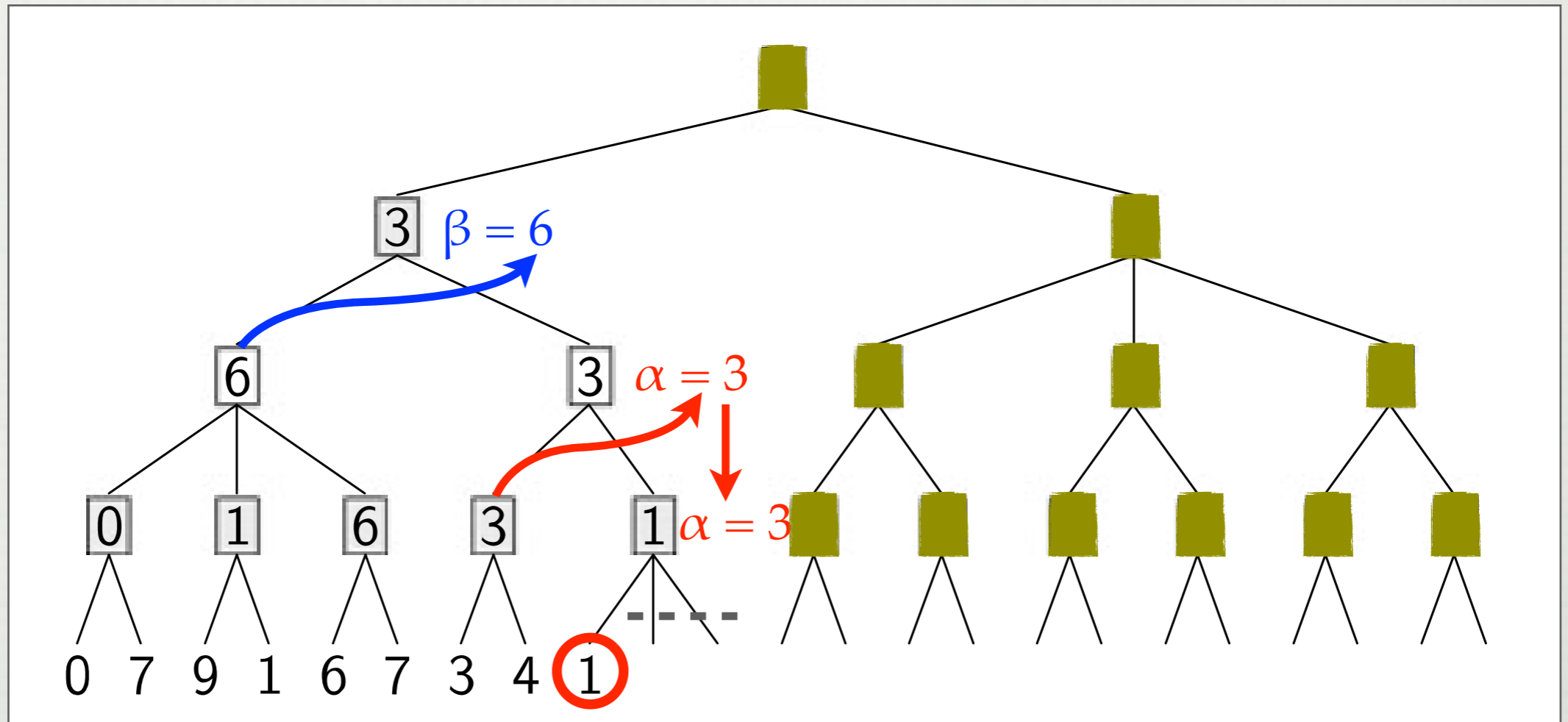
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



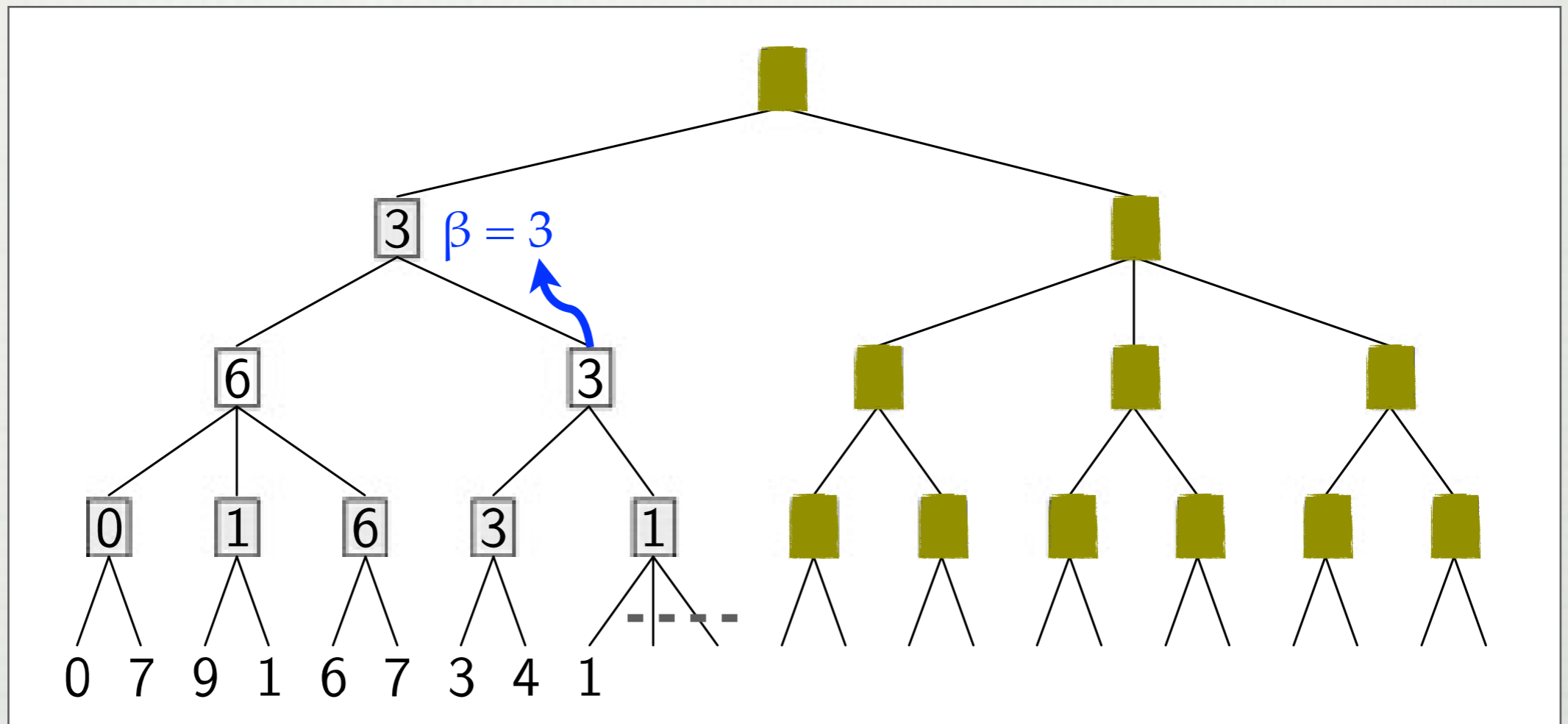
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



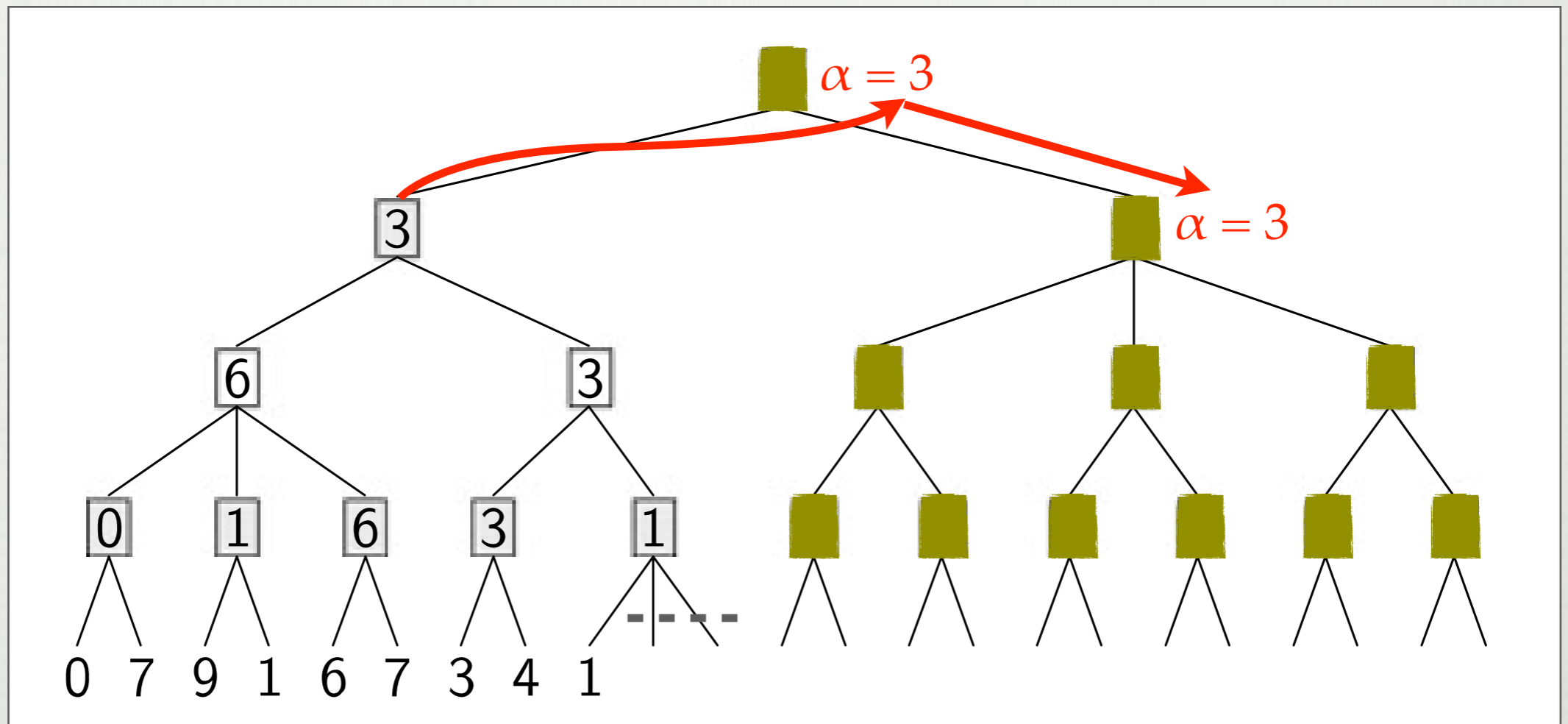
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



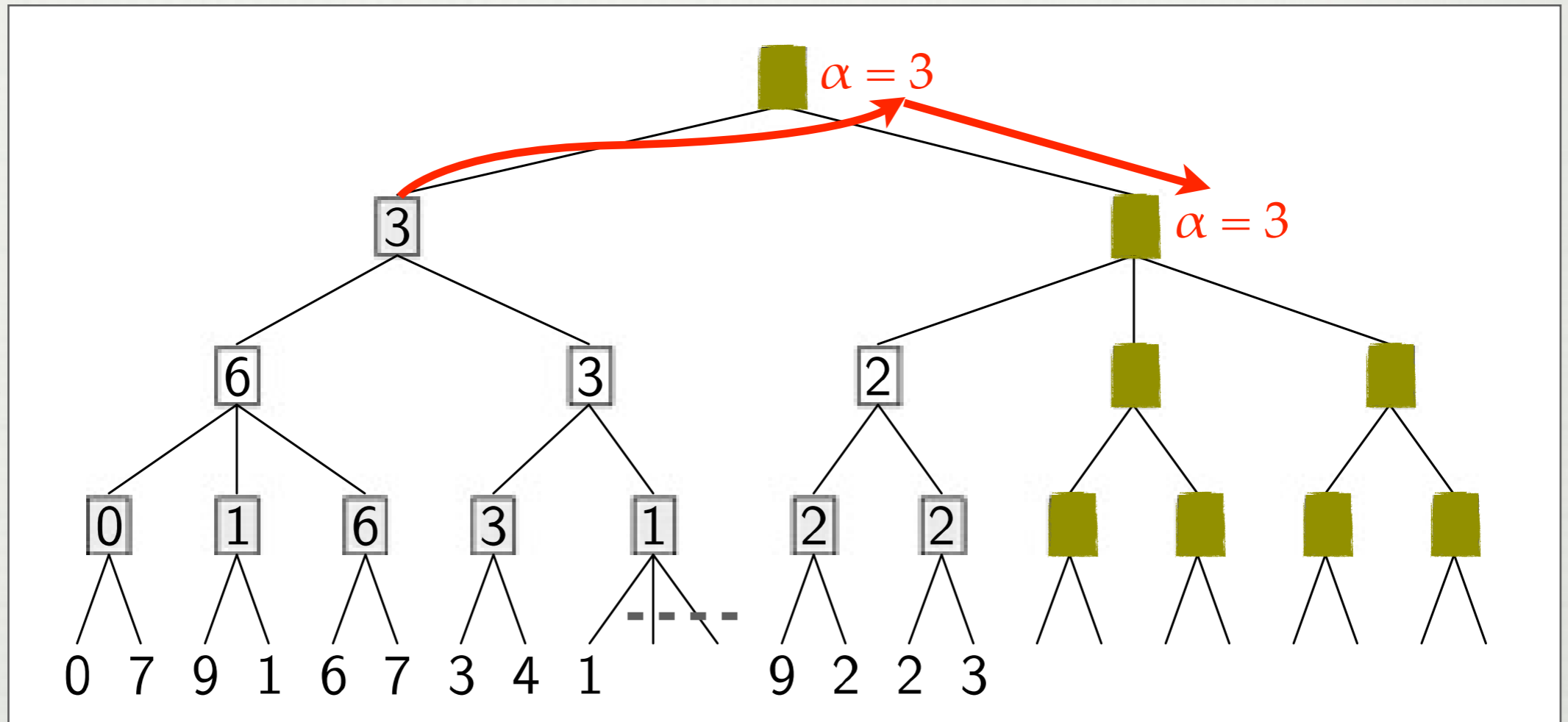
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN

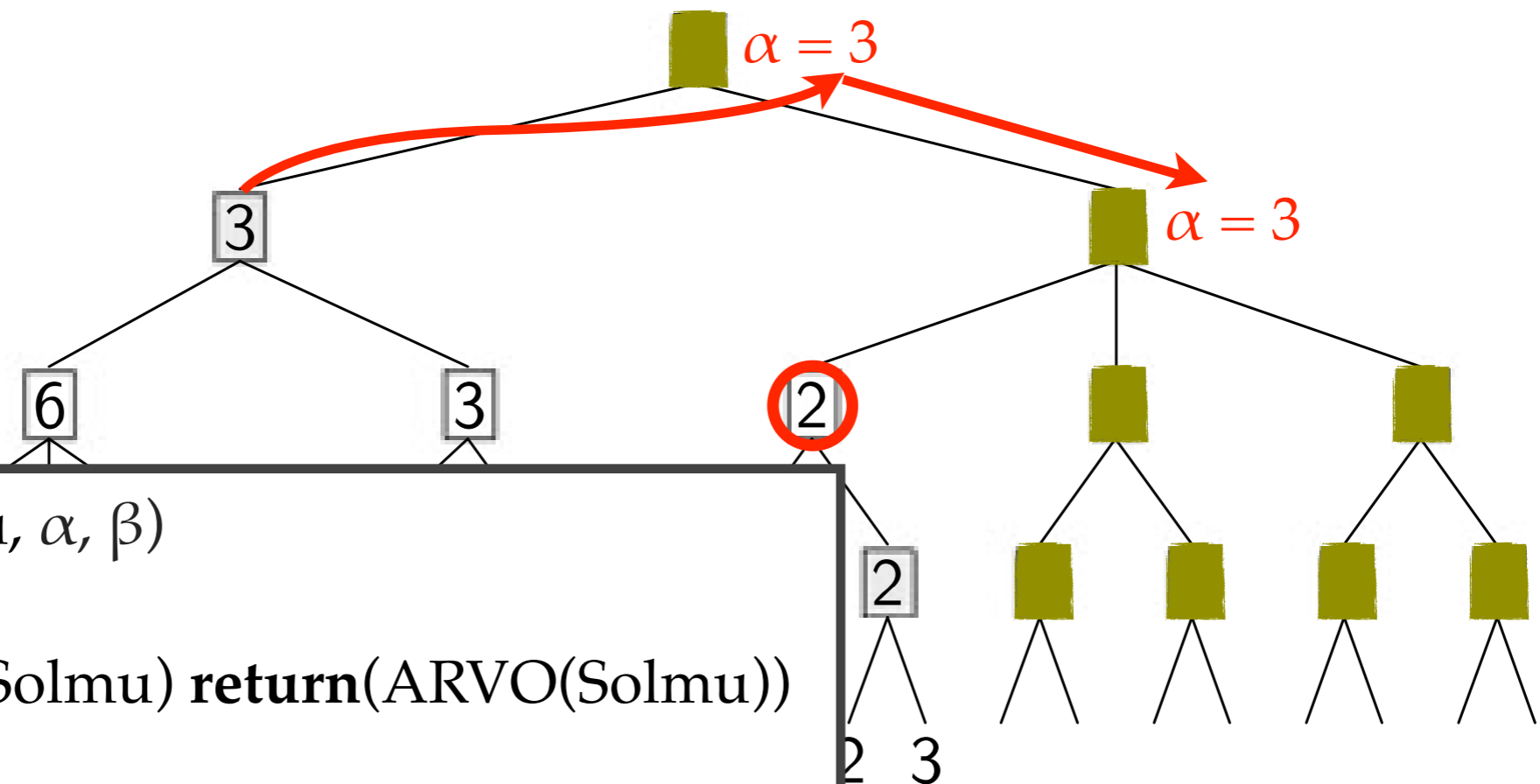


ALPHA-BETA-KARSINTA

MAX

MIN

MAX



MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = \text{MIN}(v, \text{MAX-ARVO}(\text{Lapsi}, \alpha, \beta))$

if $v \leq \alpha$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

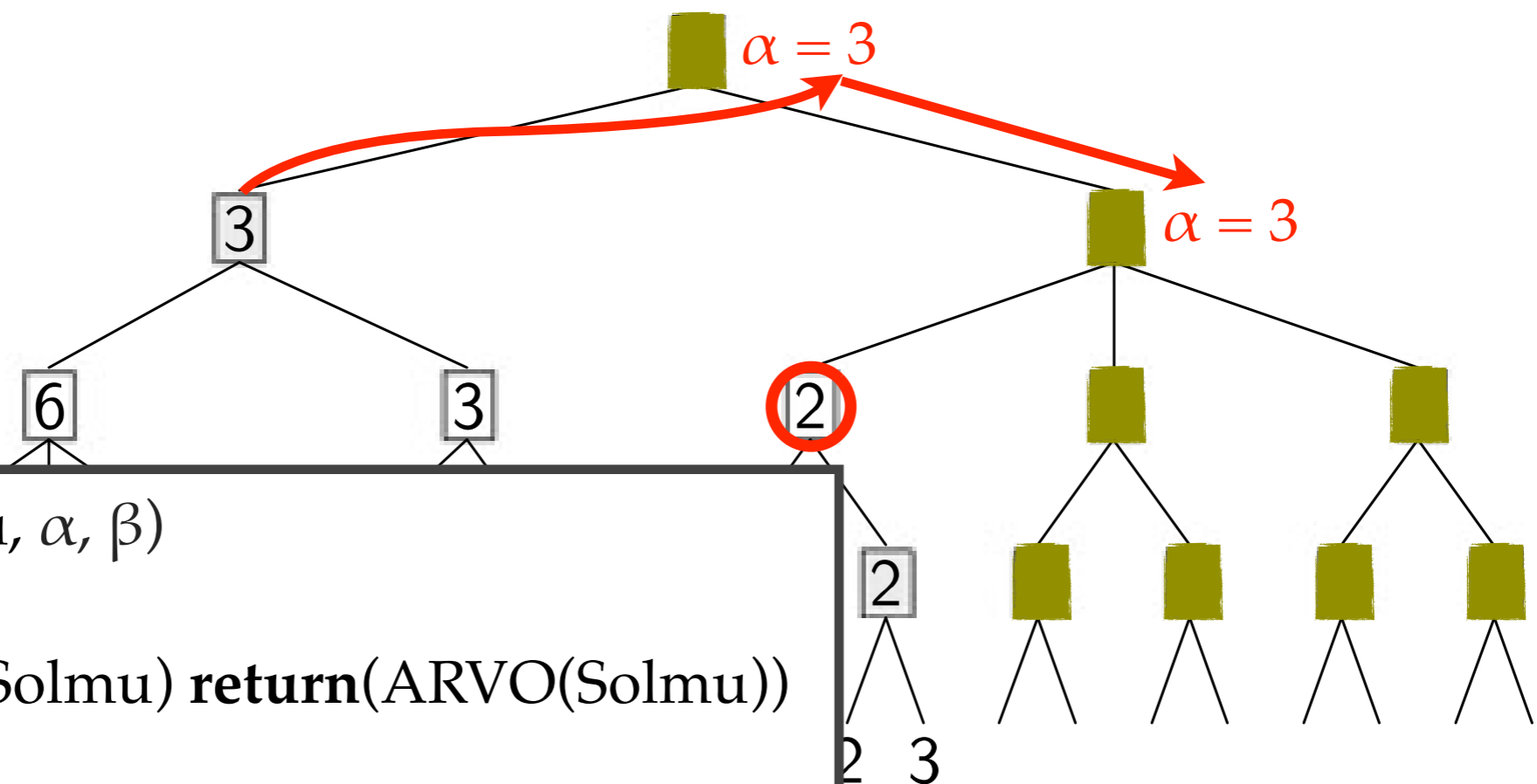
return(v)

ALPHA-BETA-KARSINTA

MAX

MIN

MAX



MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 2$

if $2 \leq 3$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

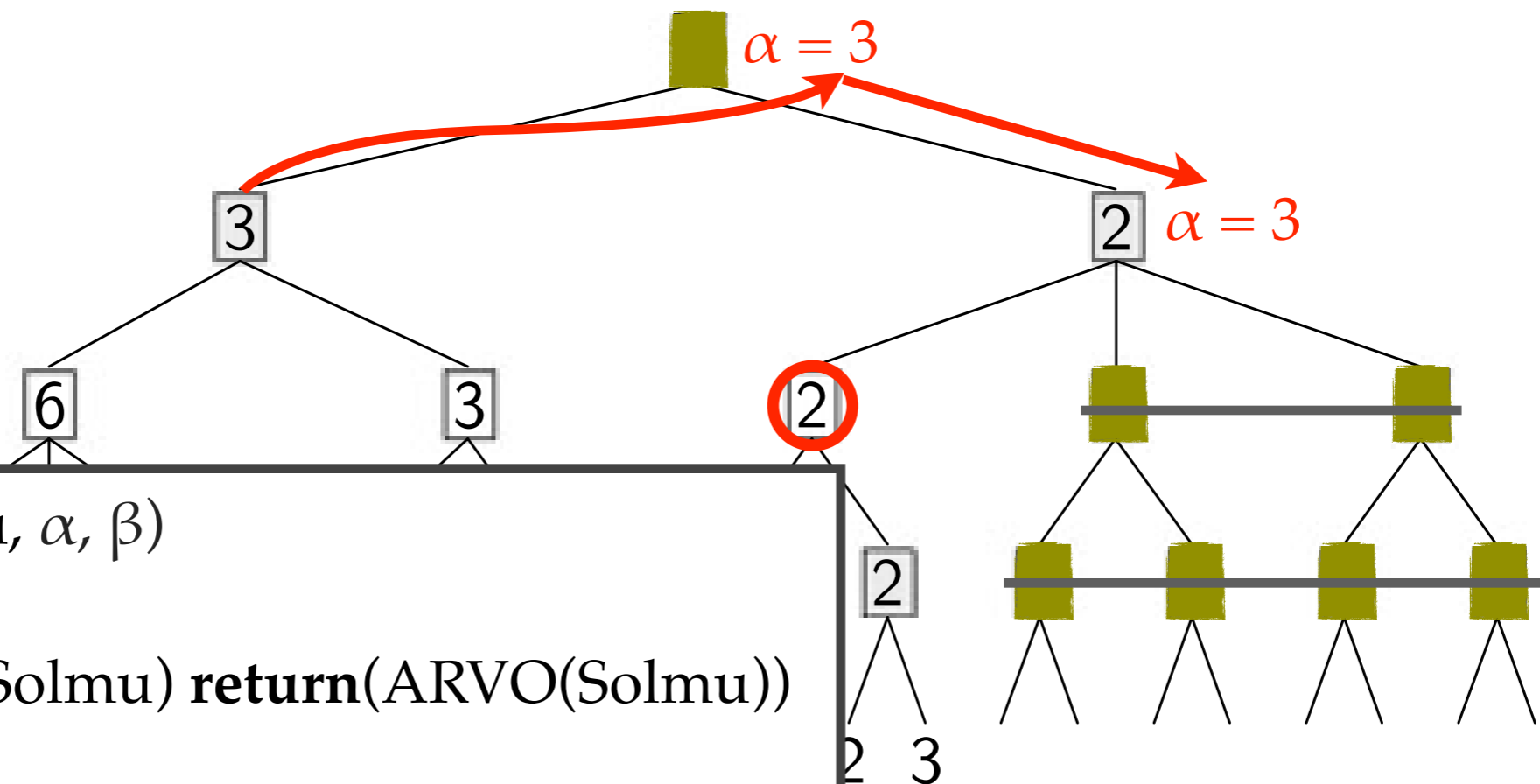
return(v)

ALPHA-BETA-KARSINTA

MAX

MIN

MAX



MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) **return**(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 2$

if $2 \leq 3$ **return**(v)

$\beta = \text{MIN}(\beta, v)$

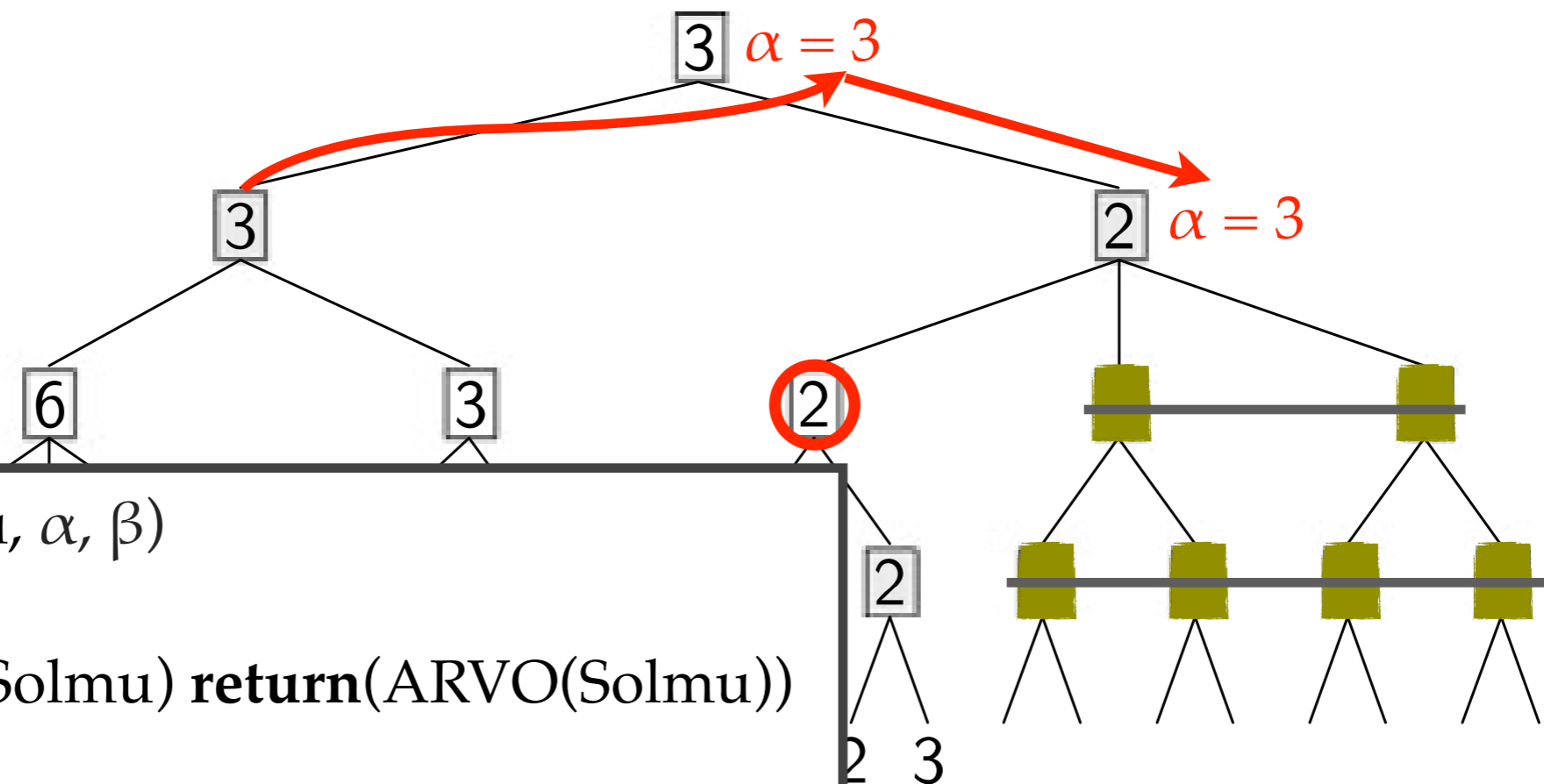
return(v)

ALPHA-BETA-KARSINTA

MAX

MIN

MAX



MIN-ARVO(Solmu, α , β)

if LOPPUTILA(Solmu) return(ARVO(Solmu))

$v = +\infty$

for each Lapsi in LAPSET(Solmu)

$v = 2$

if $2 \leq 3$ return(v)

$\beta = \text{MIN}(\beta, v)$

return(v)

ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN

