



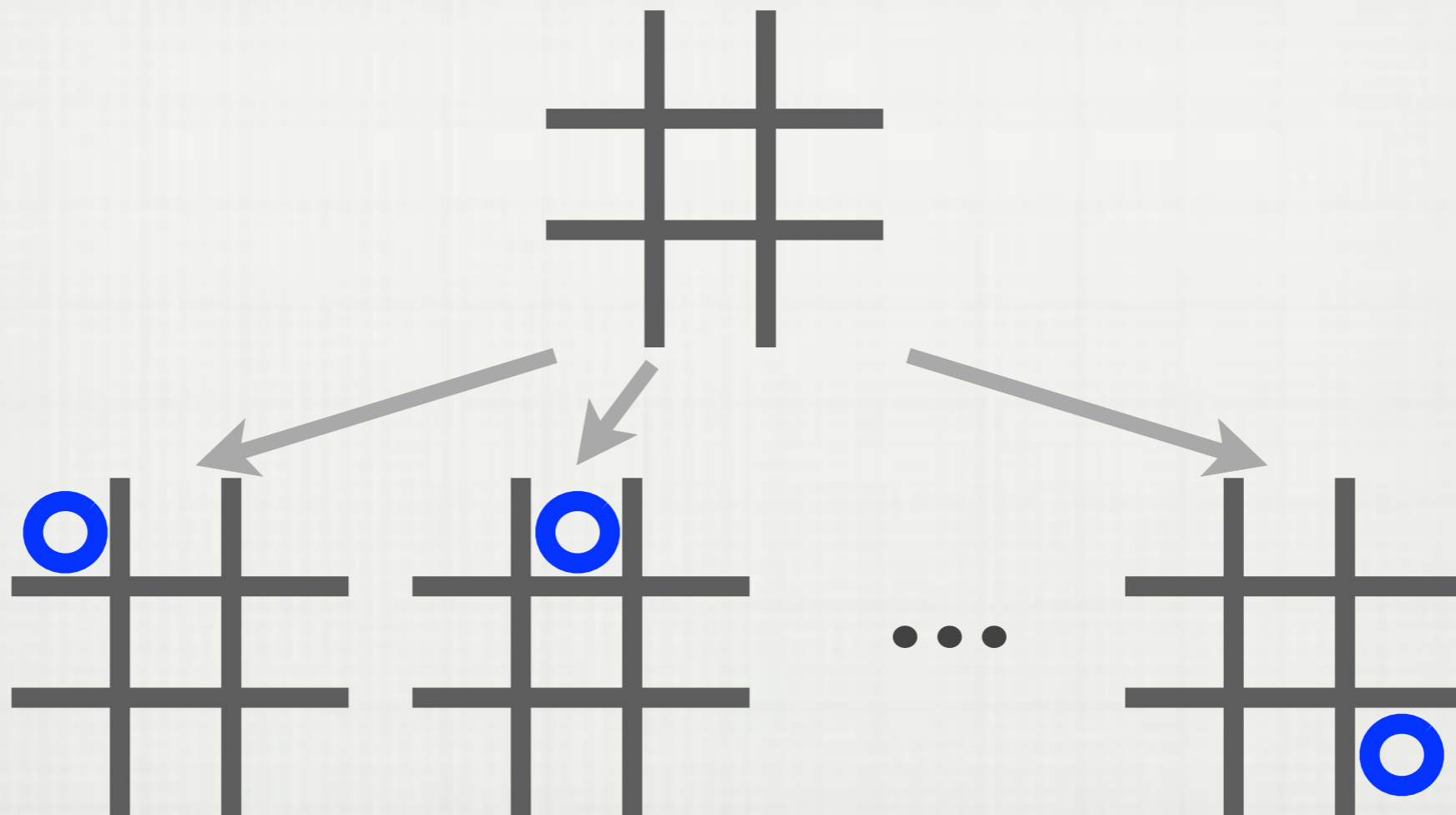
JOHDATUS TEKÖÄLYYN

TEEMU ROOS

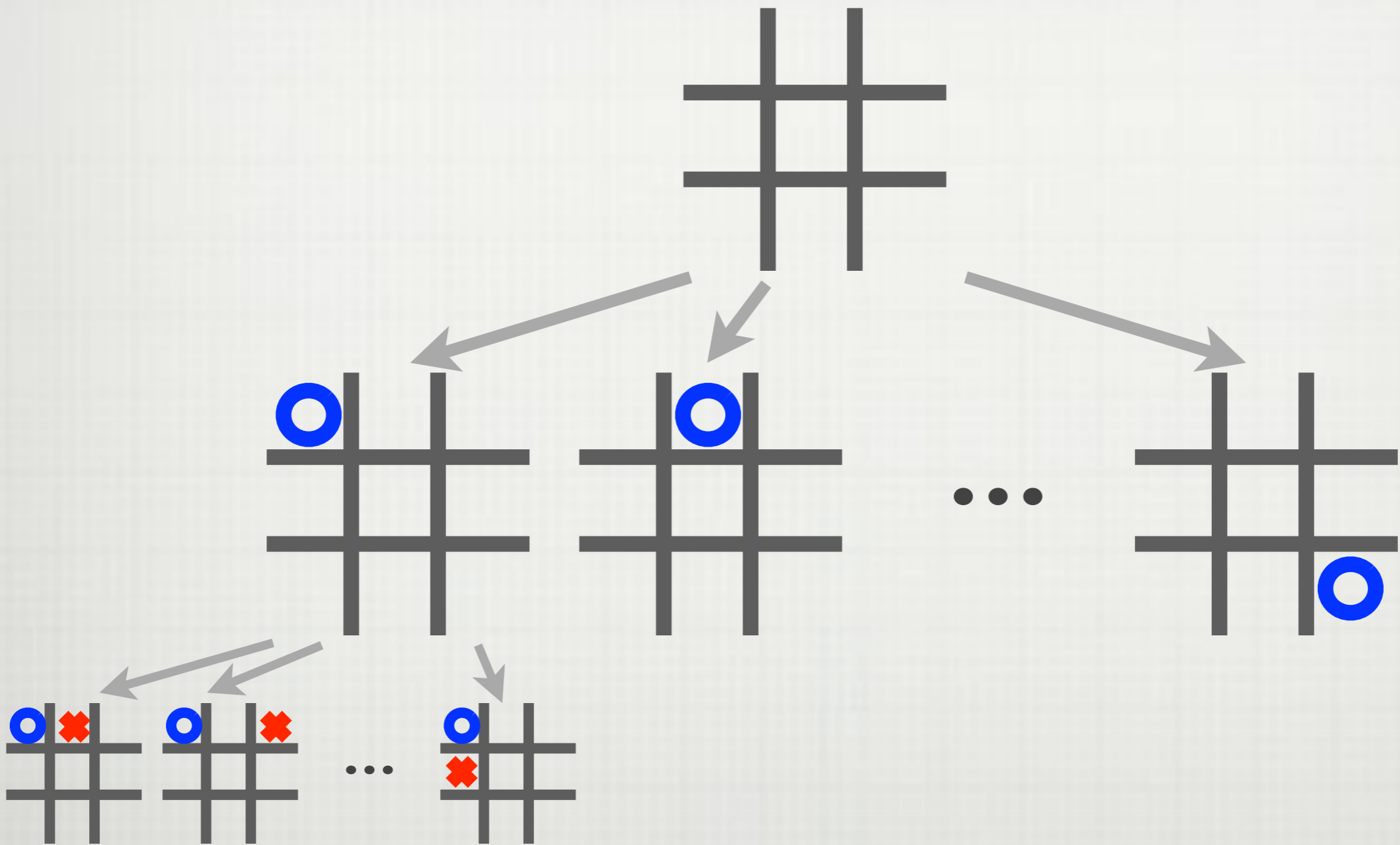


HELSINGIN YLIOPISTO

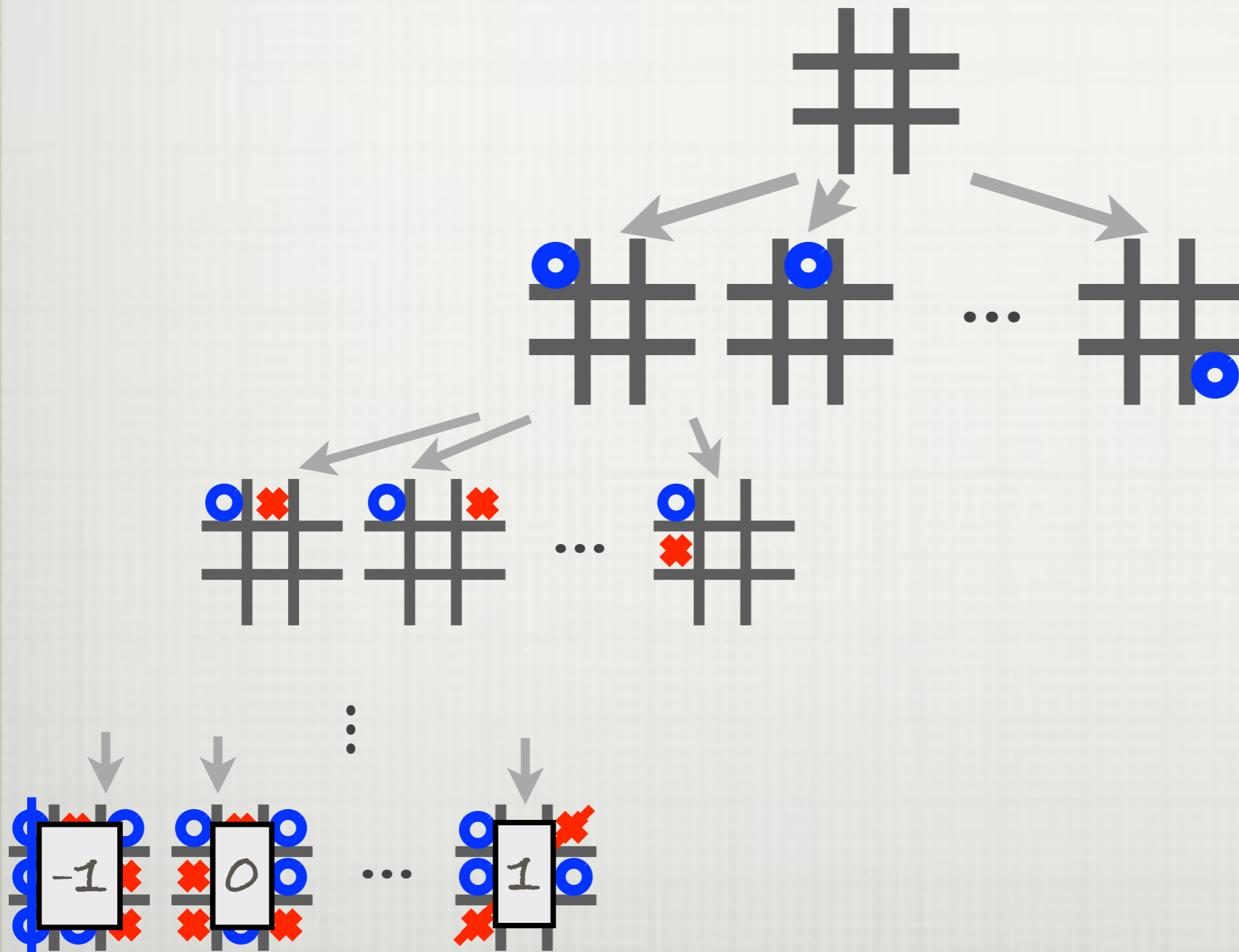
PELIPUU



PELIPUU



PELIPUU



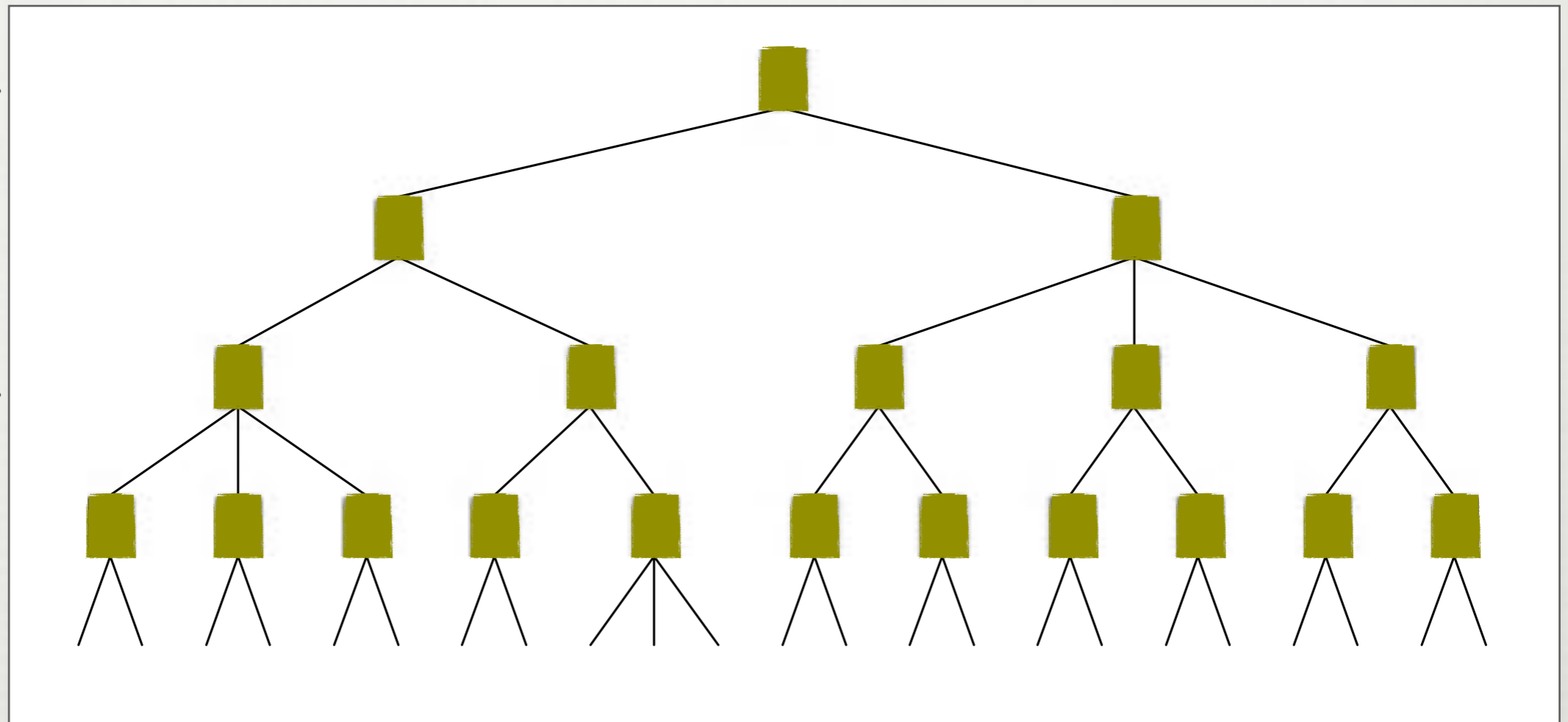
PELIPUU

MAX

MIN

MAX

MIN



PELIPUU

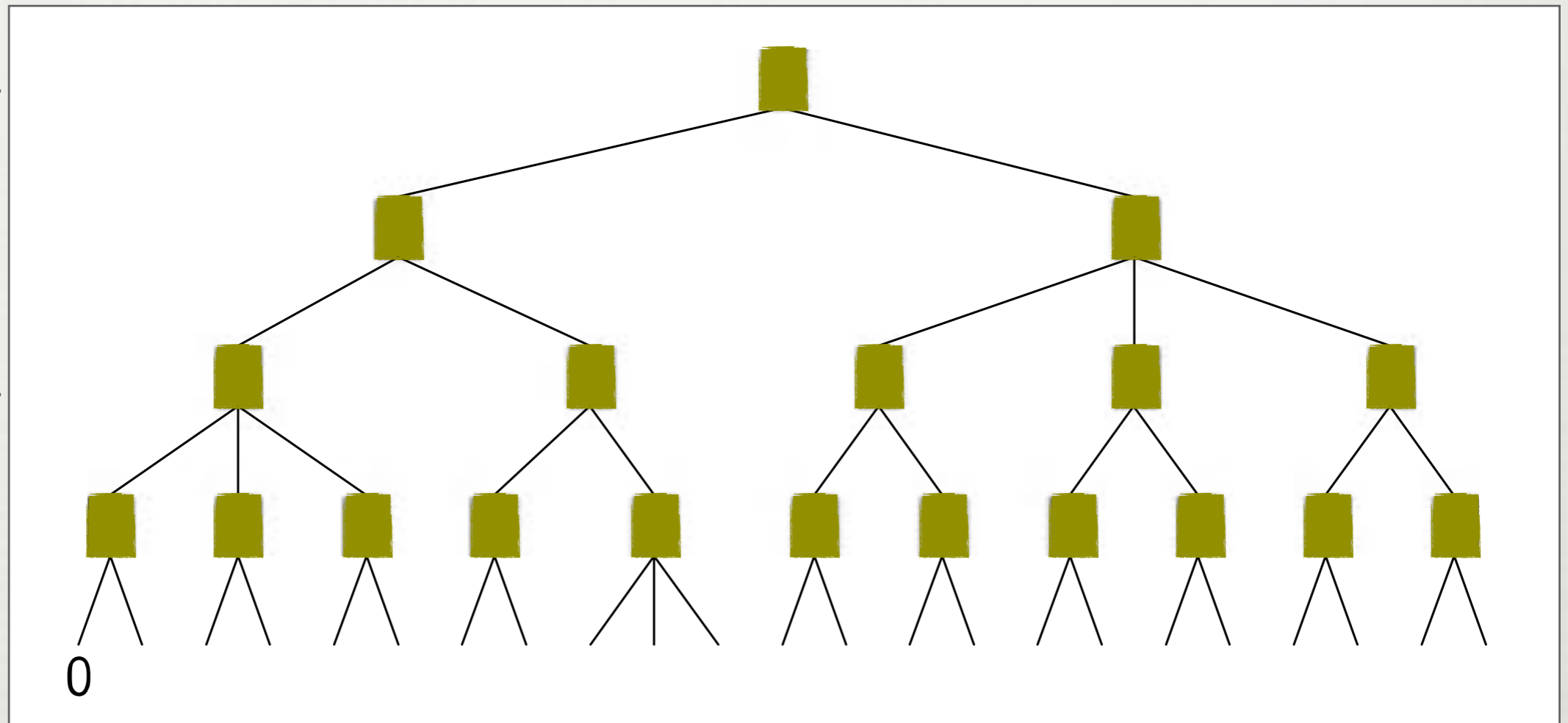
MAX

MIN

MAX

MIN

0



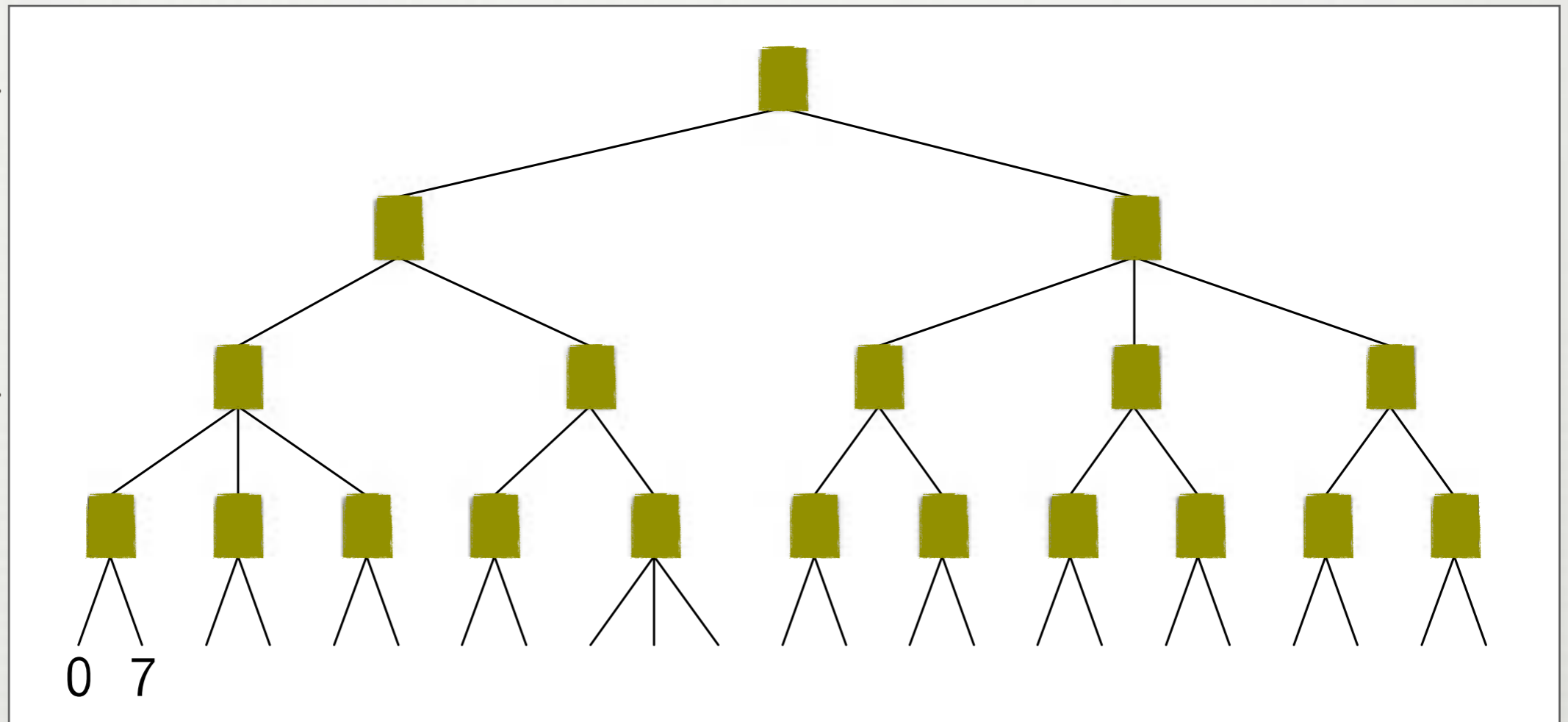
PELIPUU

MAX

MIN

MAX

MIN



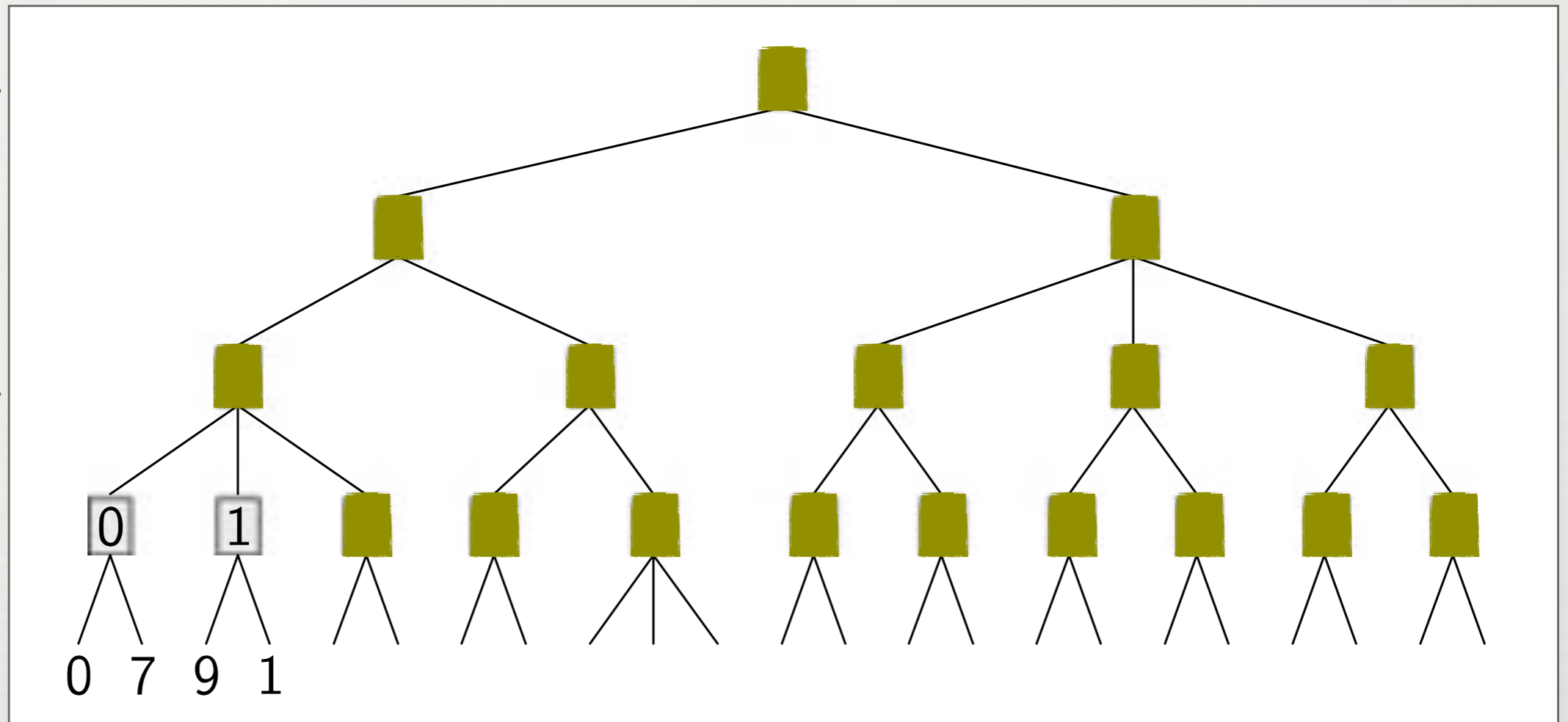
PELIPUU

MAX

MIN

MAX

MIN



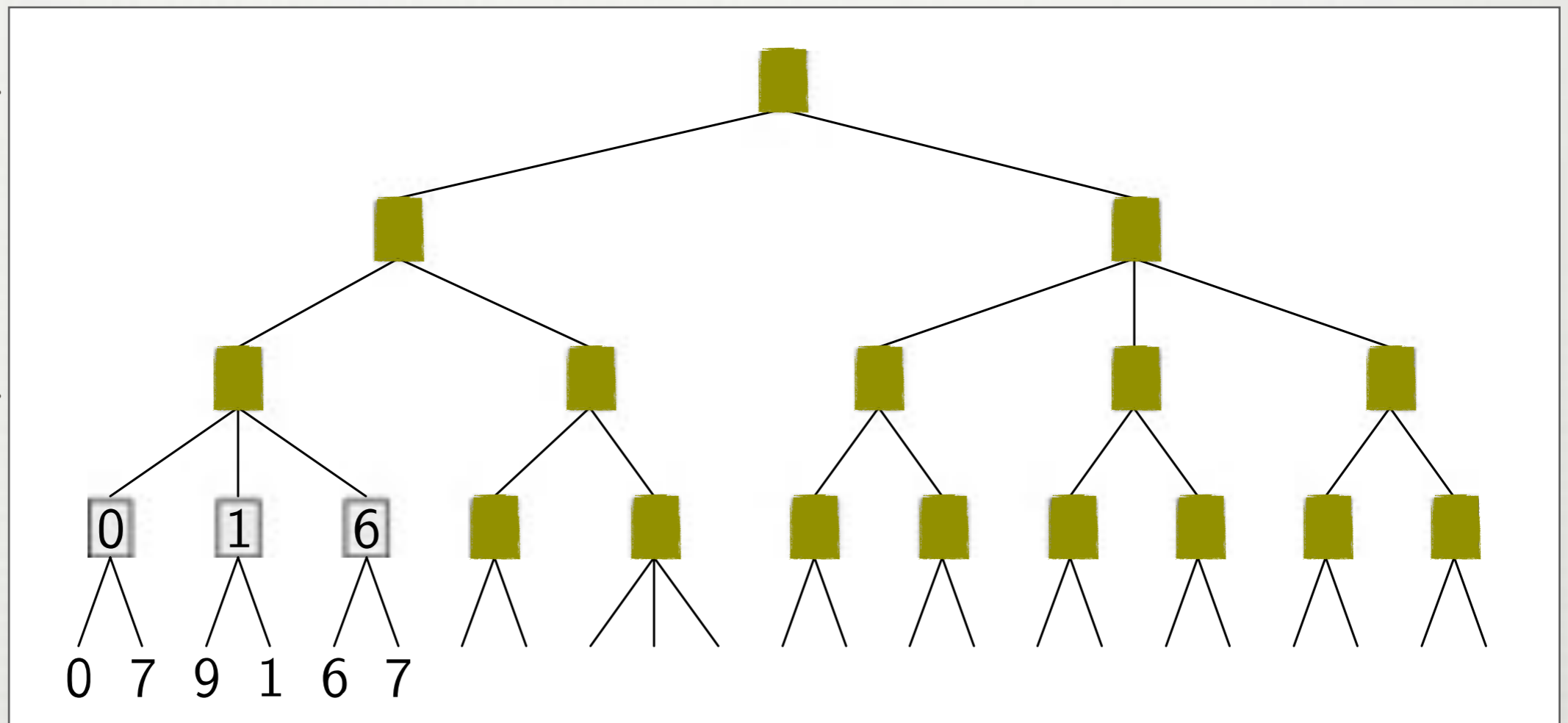
PELIPUU

MAX

MIN

MAX

MIN



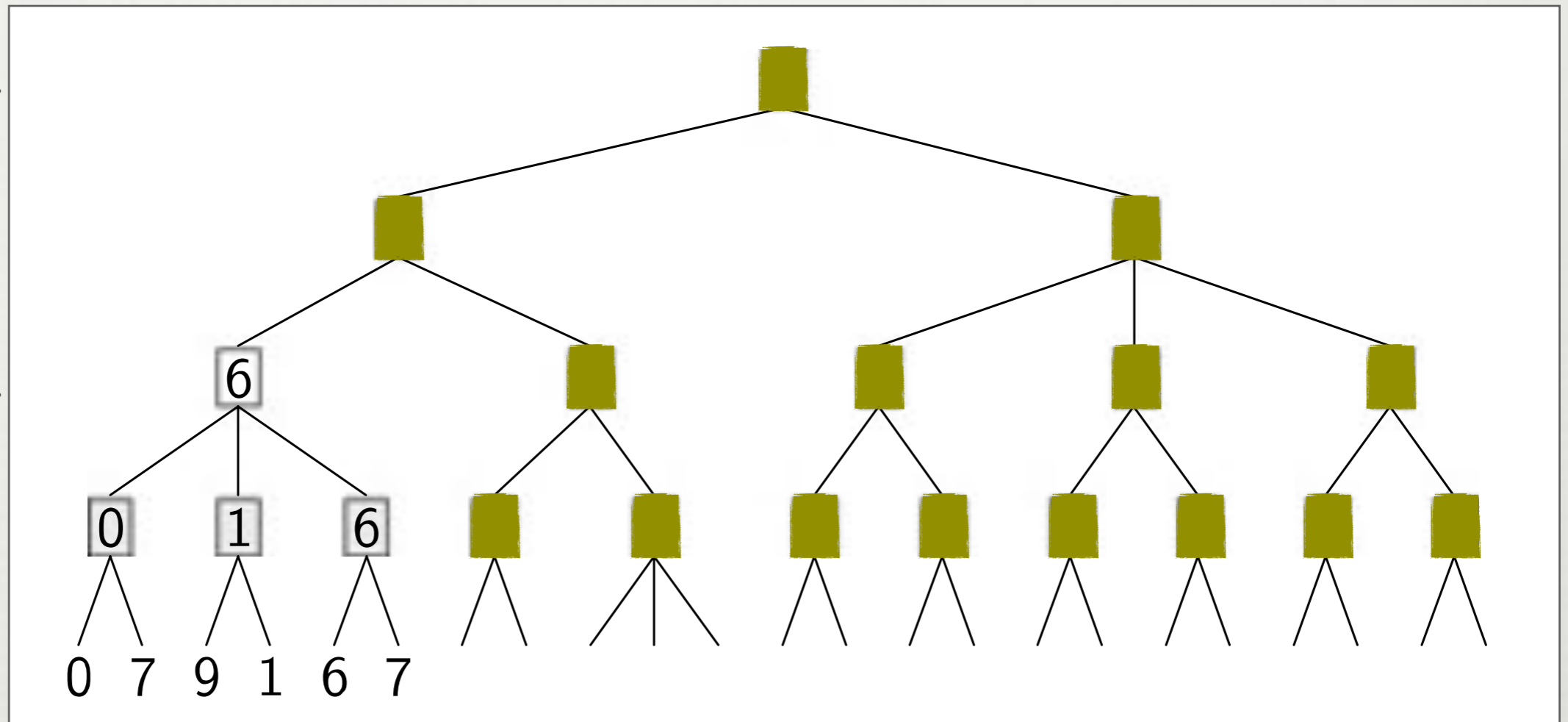
PELIPUU

MAX

MIN

MAX

MIN



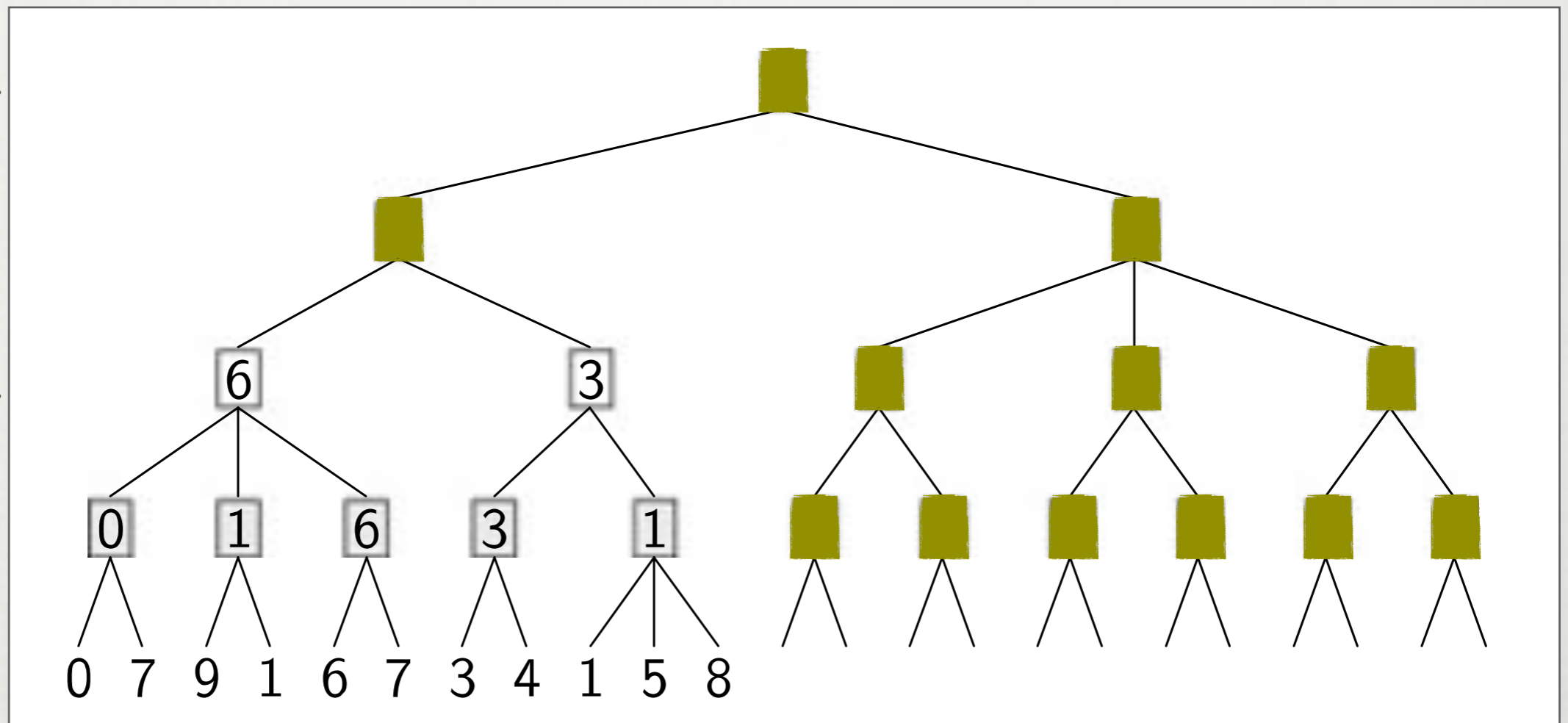
PELIPUU

MAX

MIN

MAX

MIN



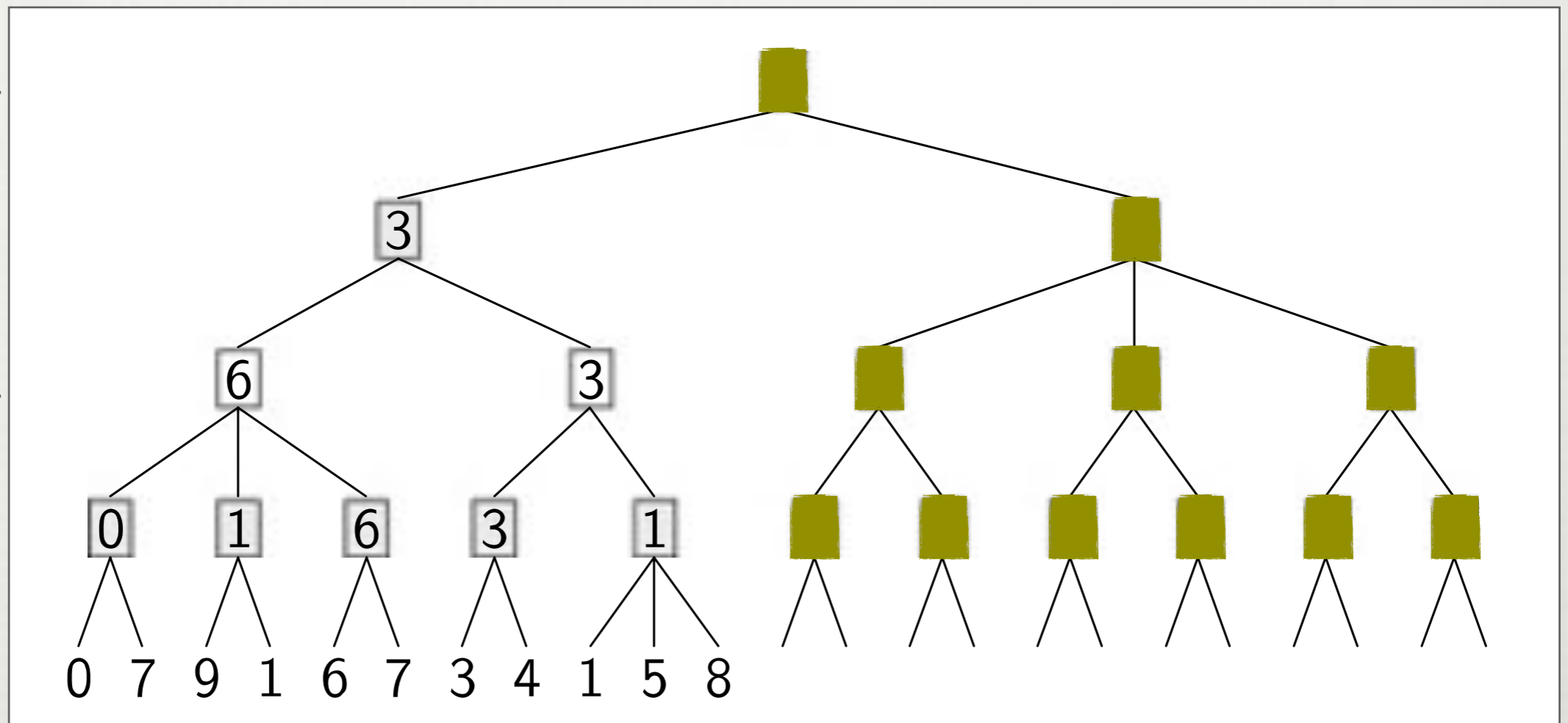
PELIPUU

MAX

MIN

MAX

MIN



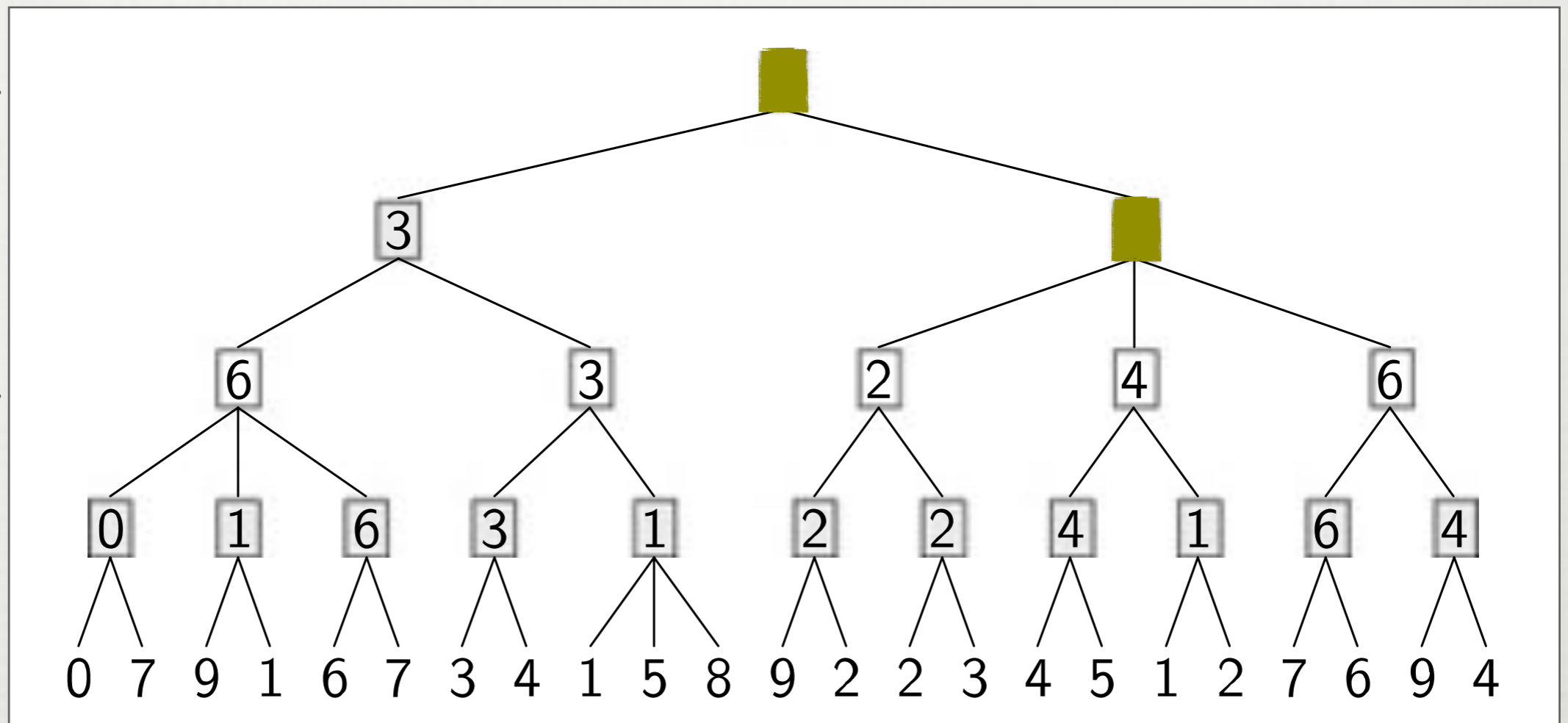
PELIPUU

MAX

MIN

MAX

MIN



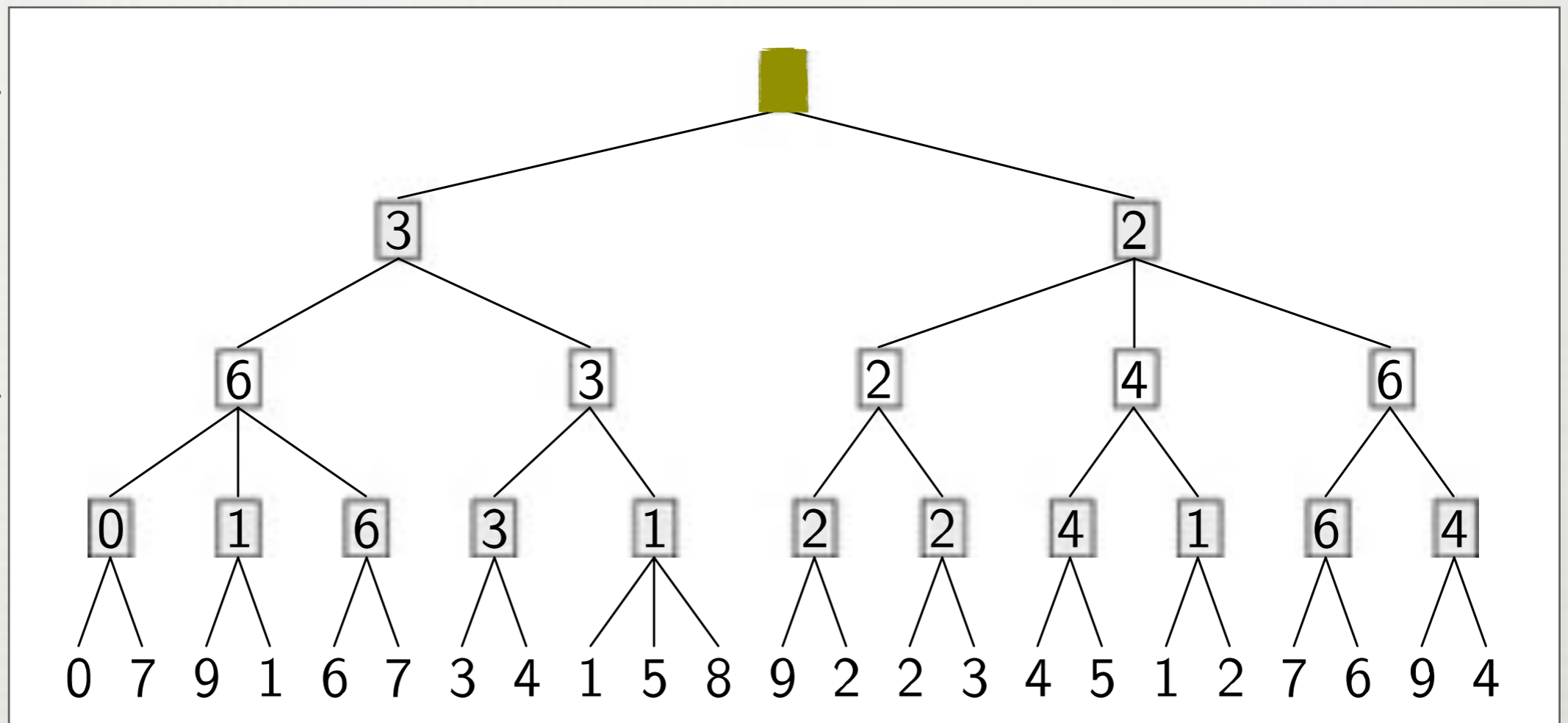
PELIPUU

MAX

MIN

MAX

MIN



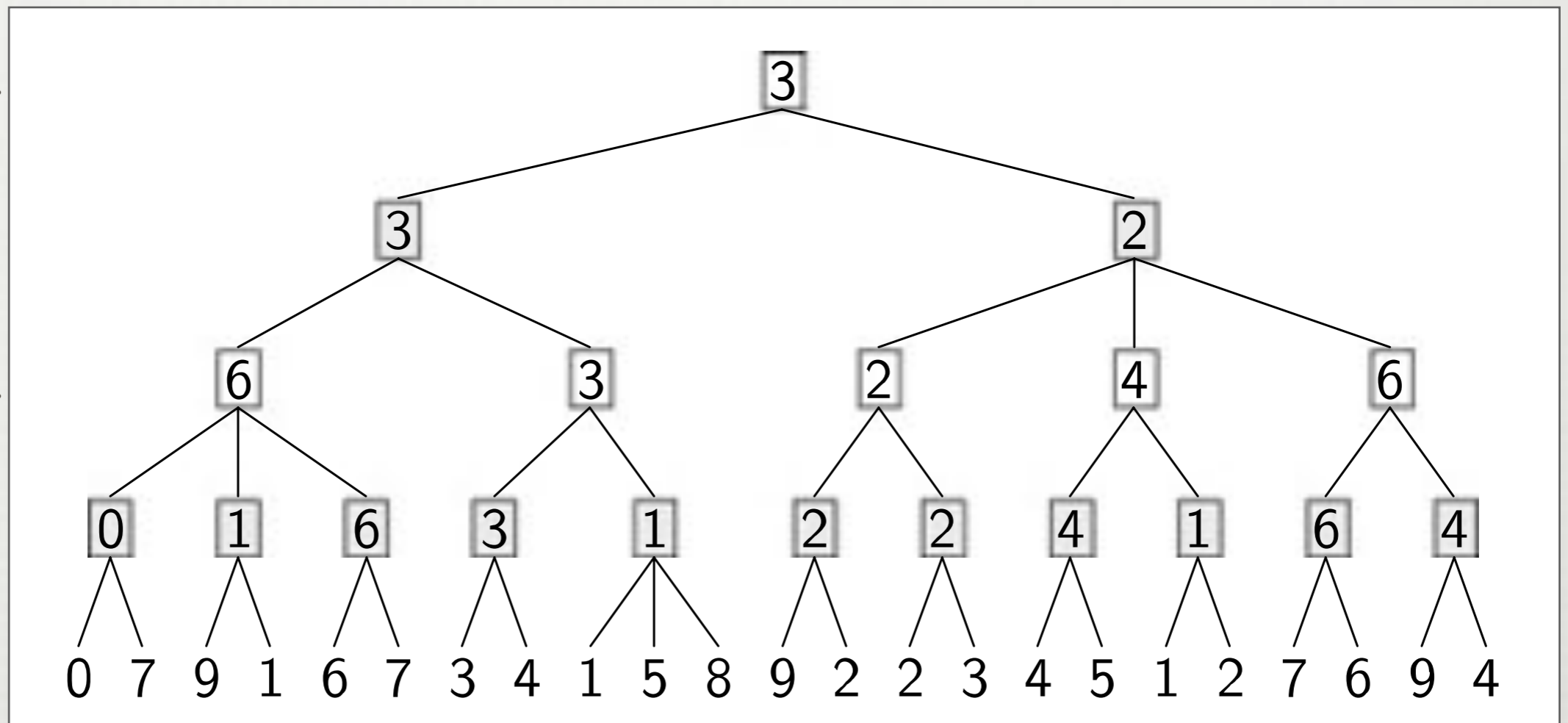
PELIPUU

MAX

MIN

MAX

MIN



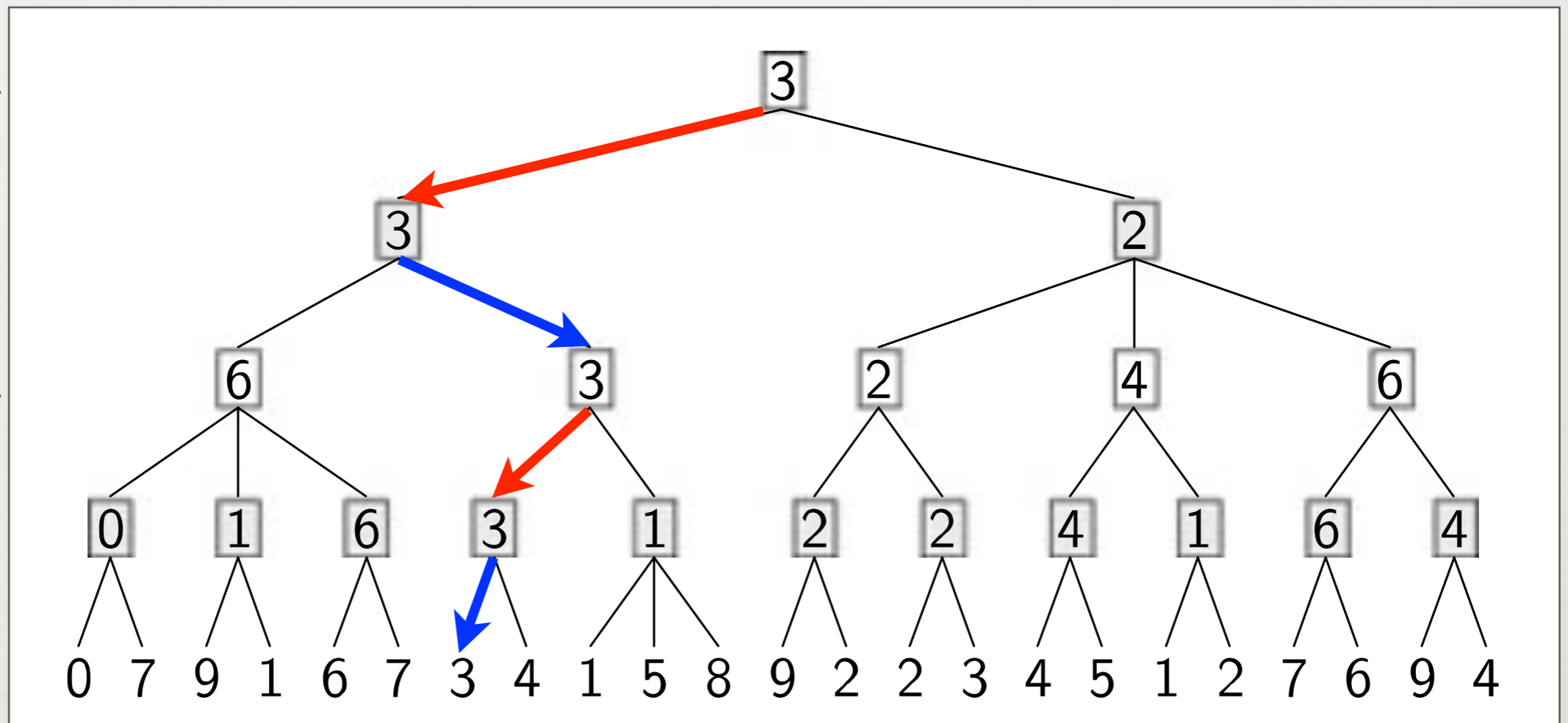
PELIPUU

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

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 Demonstration

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 search type menu item

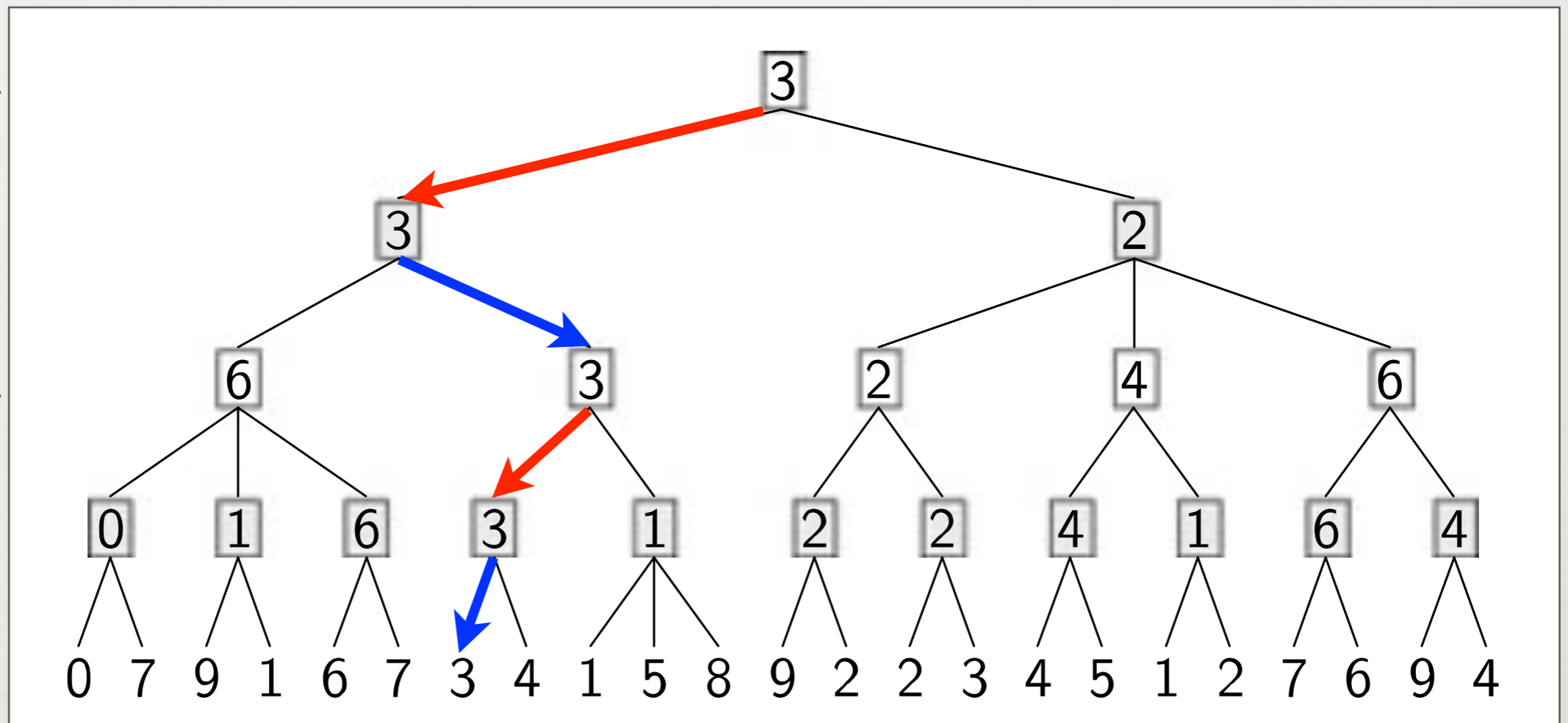
PELIPUU

MAX

MIN

MAX

MIN



ALPHA-BETA-KARSINTA

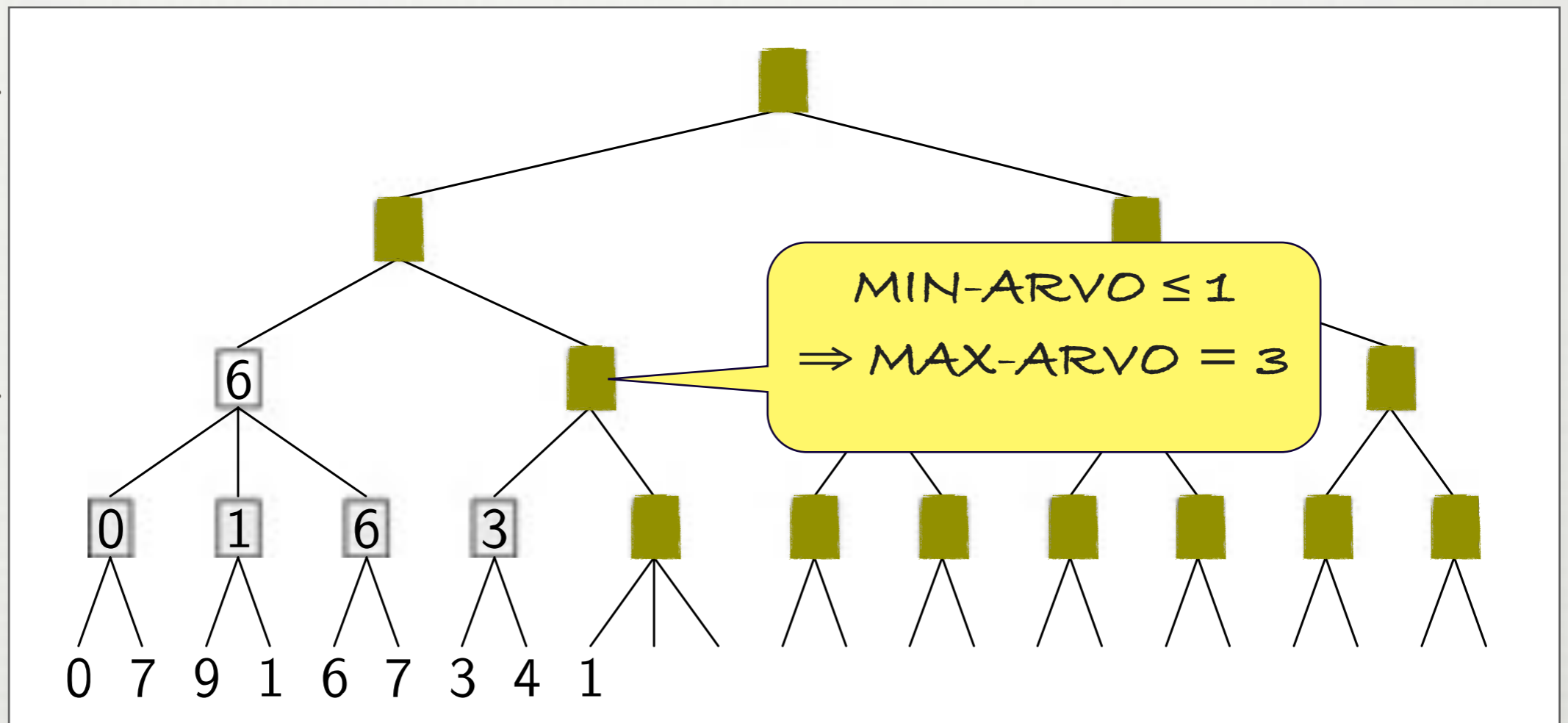
PELIPUU

MAX

MIN

MAX

MIN



ALPHA-BETA-KARSINTA

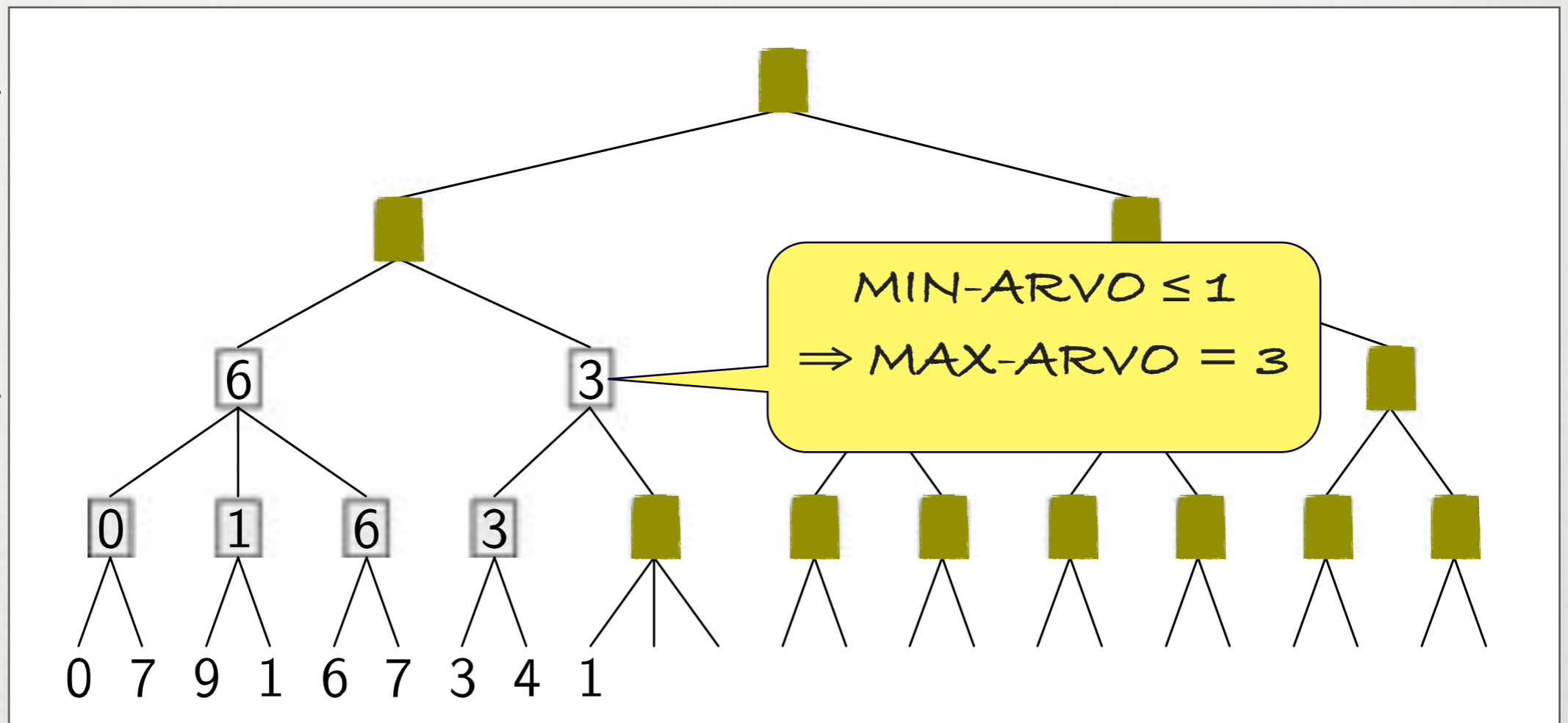
PELIPUU

MAX

MIN

MAX

MIN



ALPHA-BETA-KARSINTA

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)

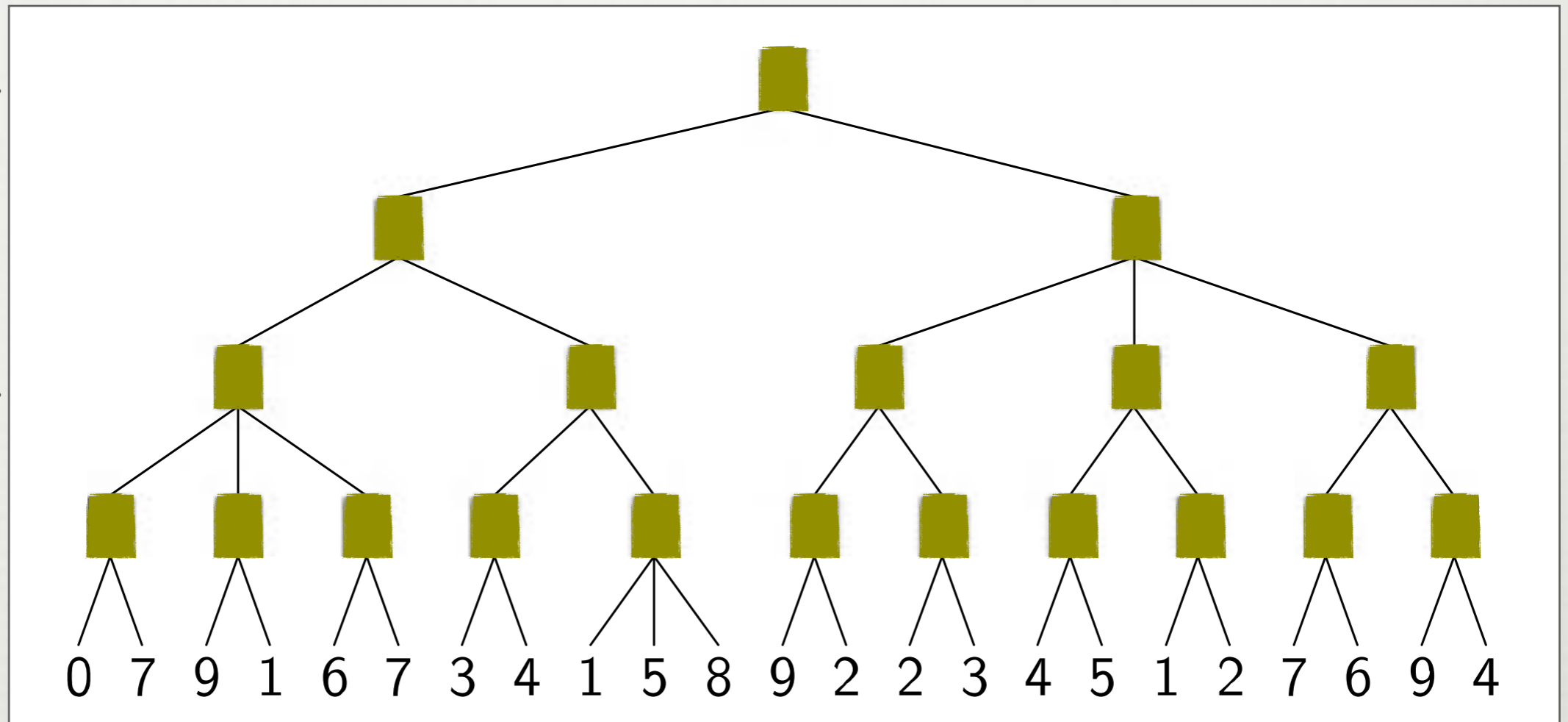
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



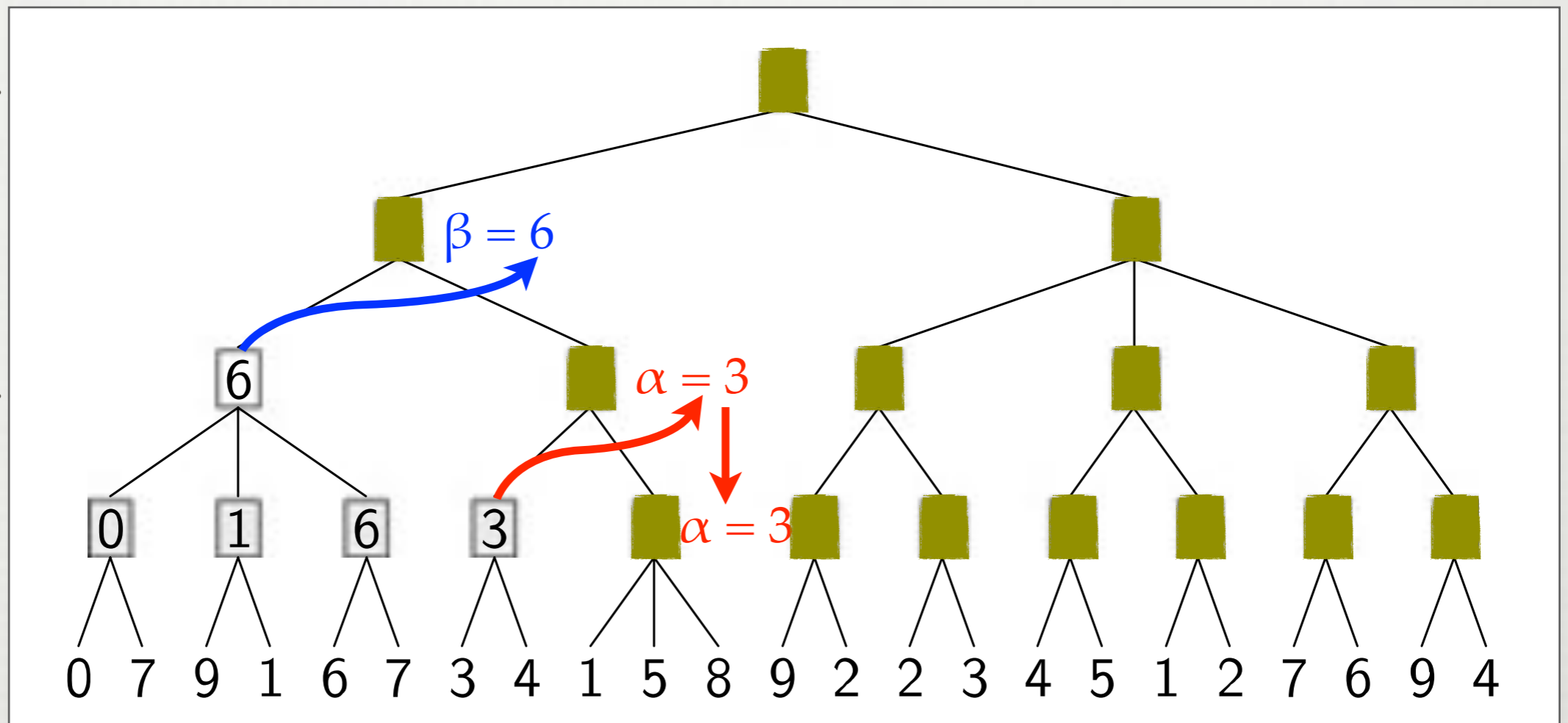
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



ALPHA

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

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

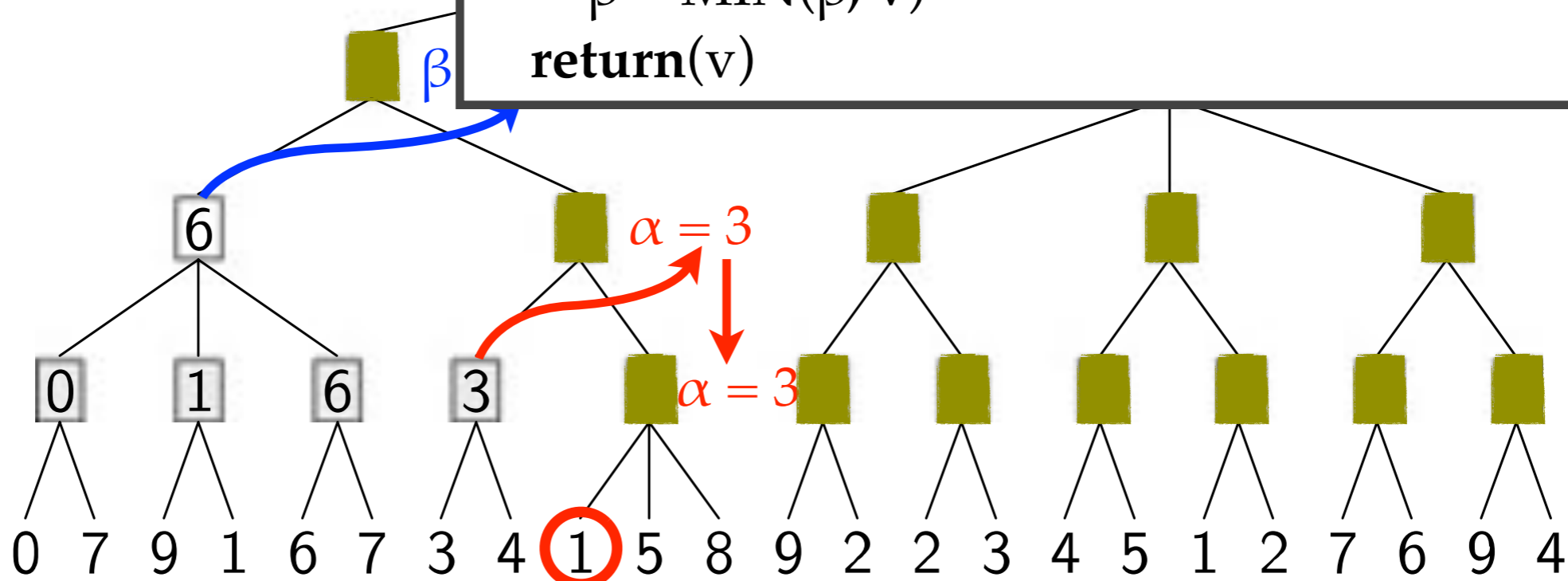
return(v)

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 $1 \leq 3$ return v

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

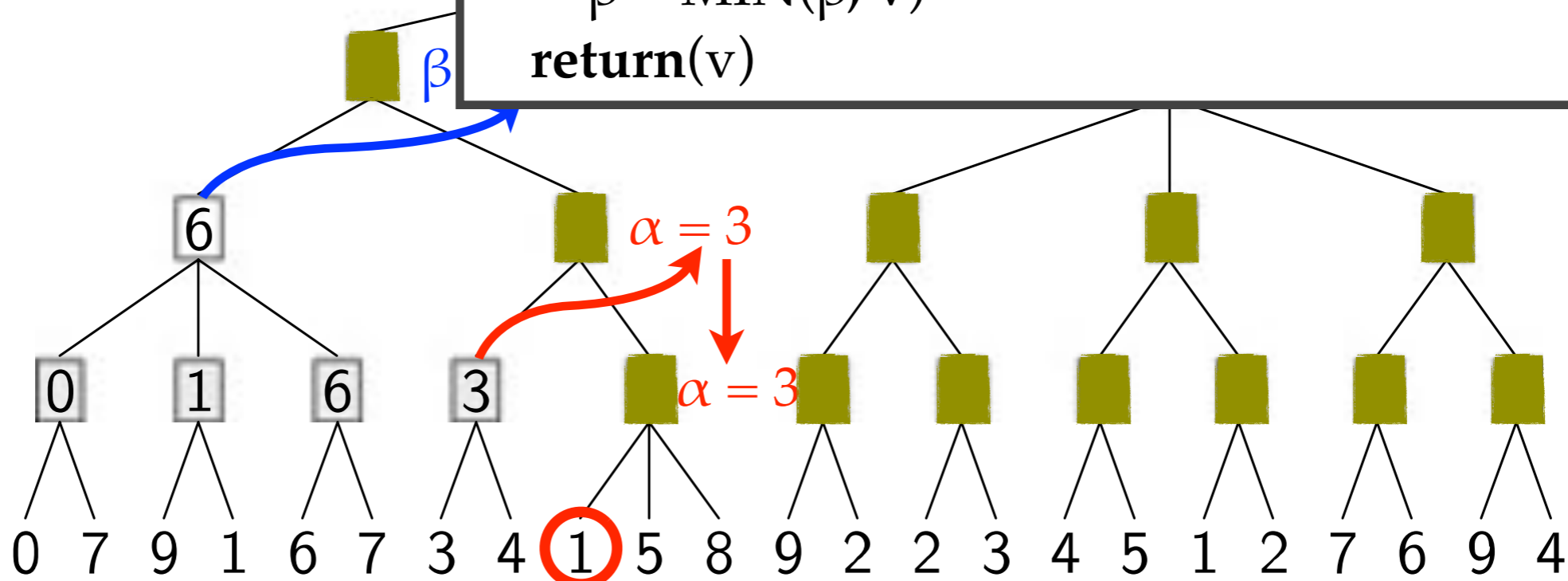
return(v)

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

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

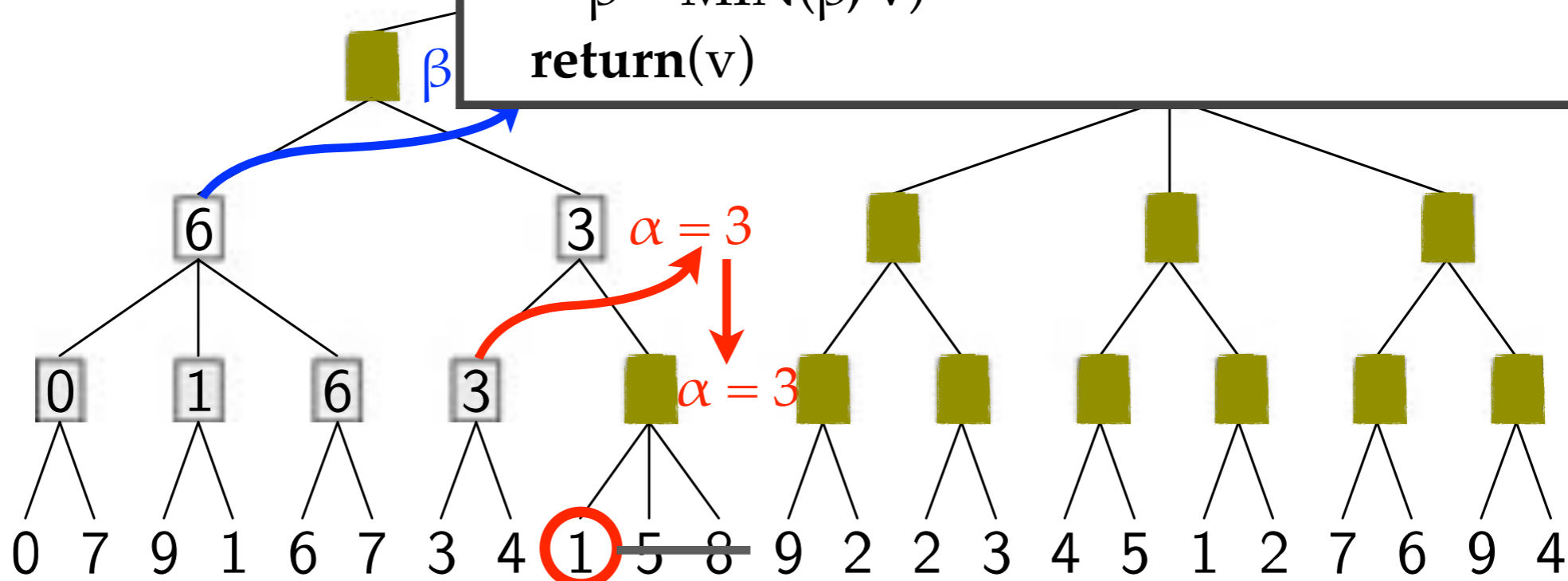
return(v)

MAX

MIN

MAX

MIN



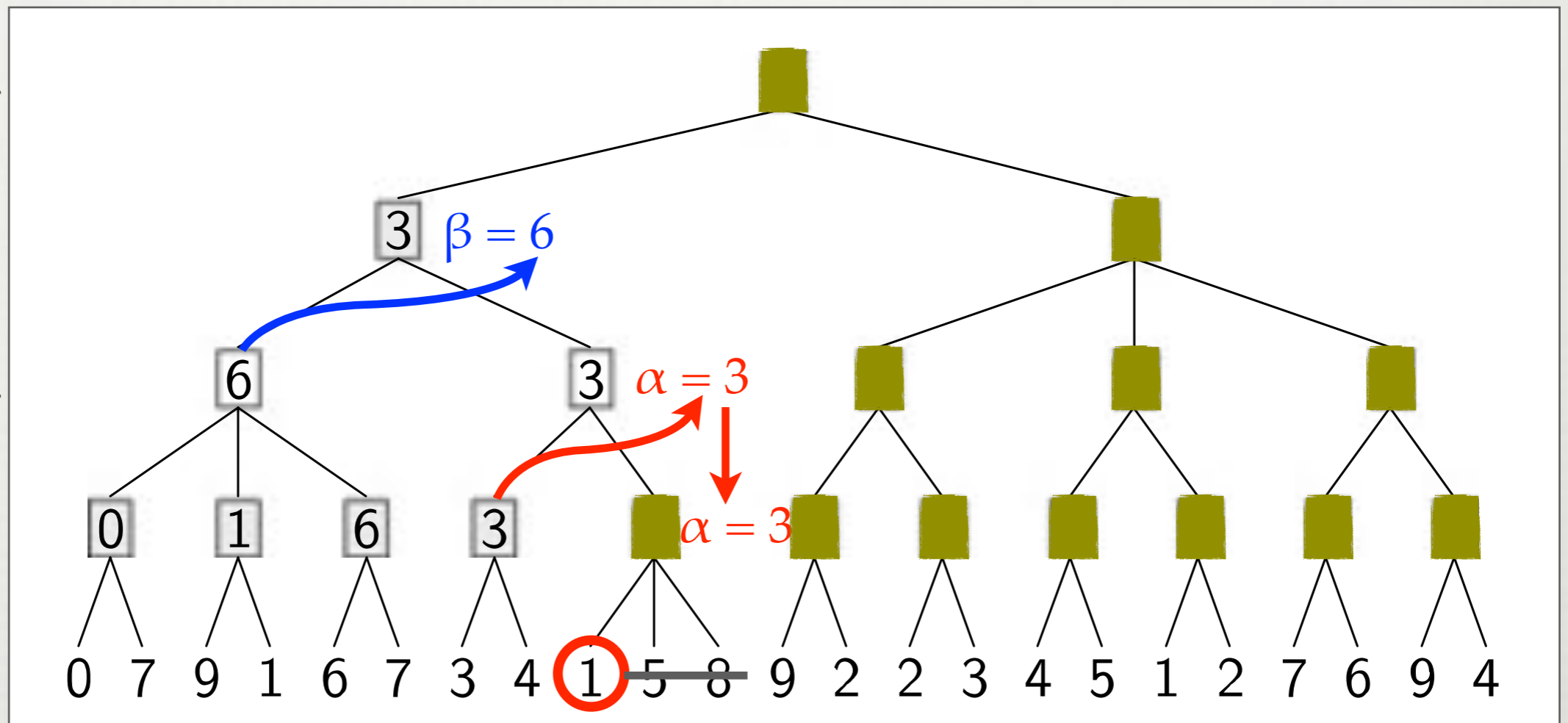
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN



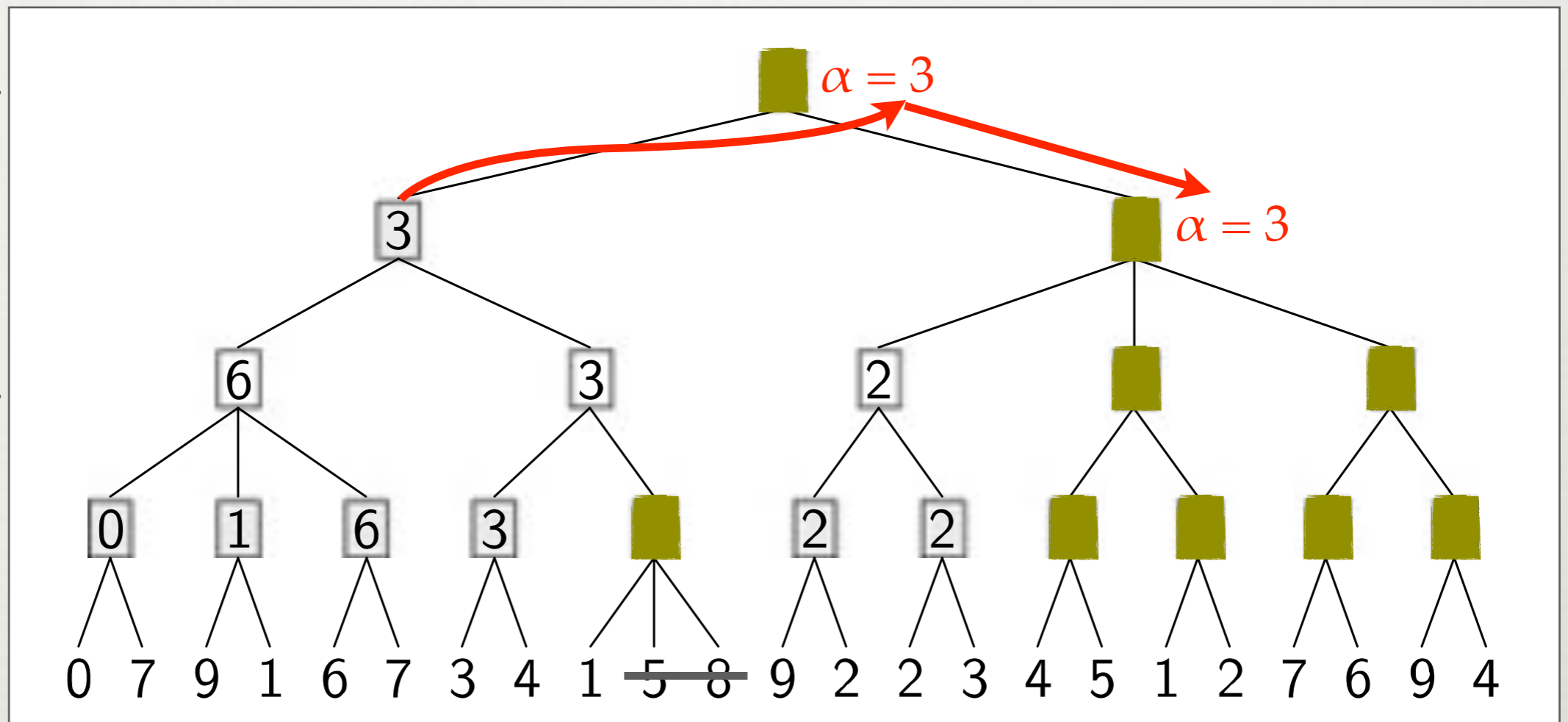
ALPHA-BETA-KARSINTA

MAX

MIN

MAX

MIN

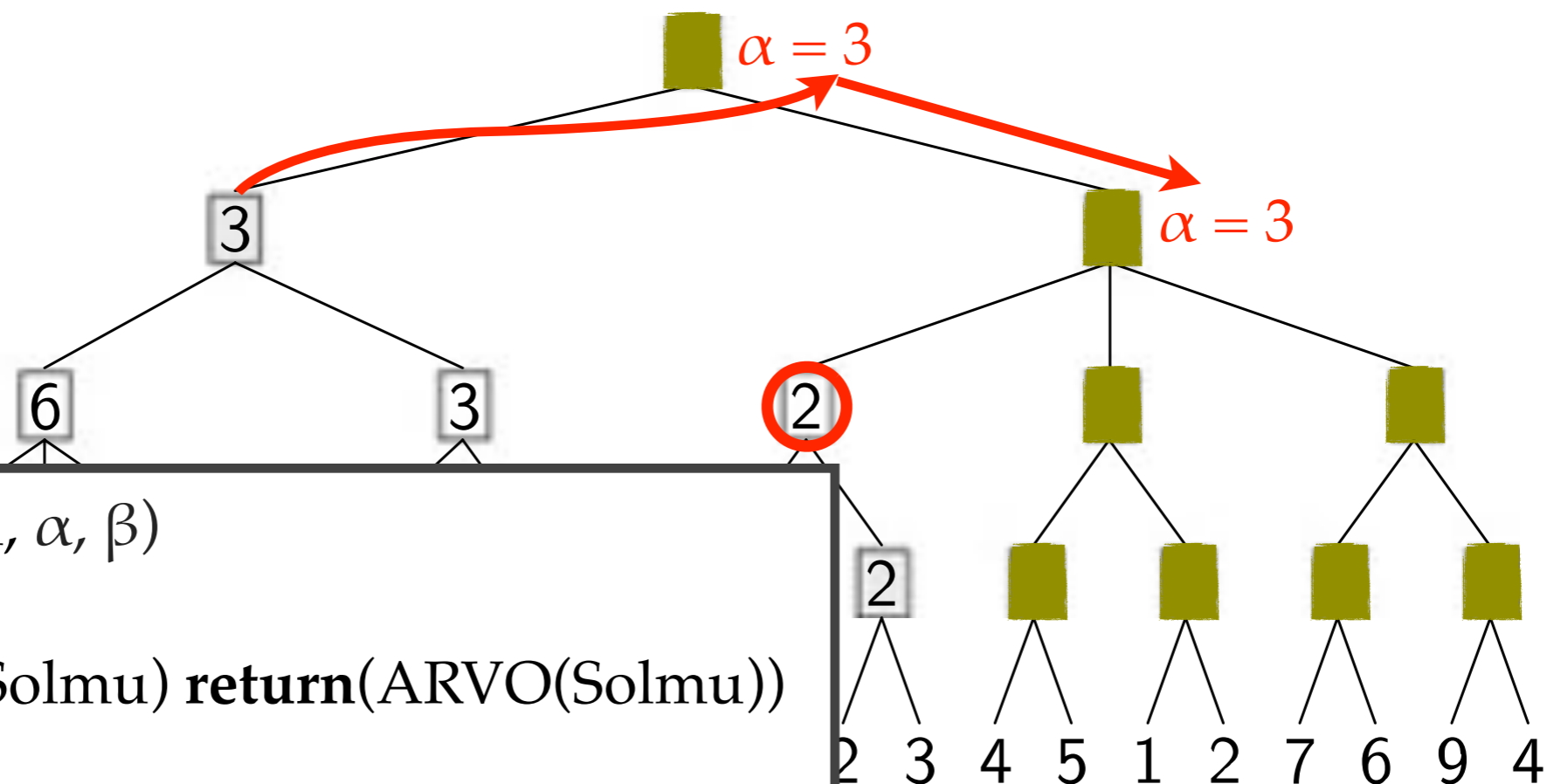


ALPHA-BETA-KARSINTA

MAX

MIN

MAX



MIN-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

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

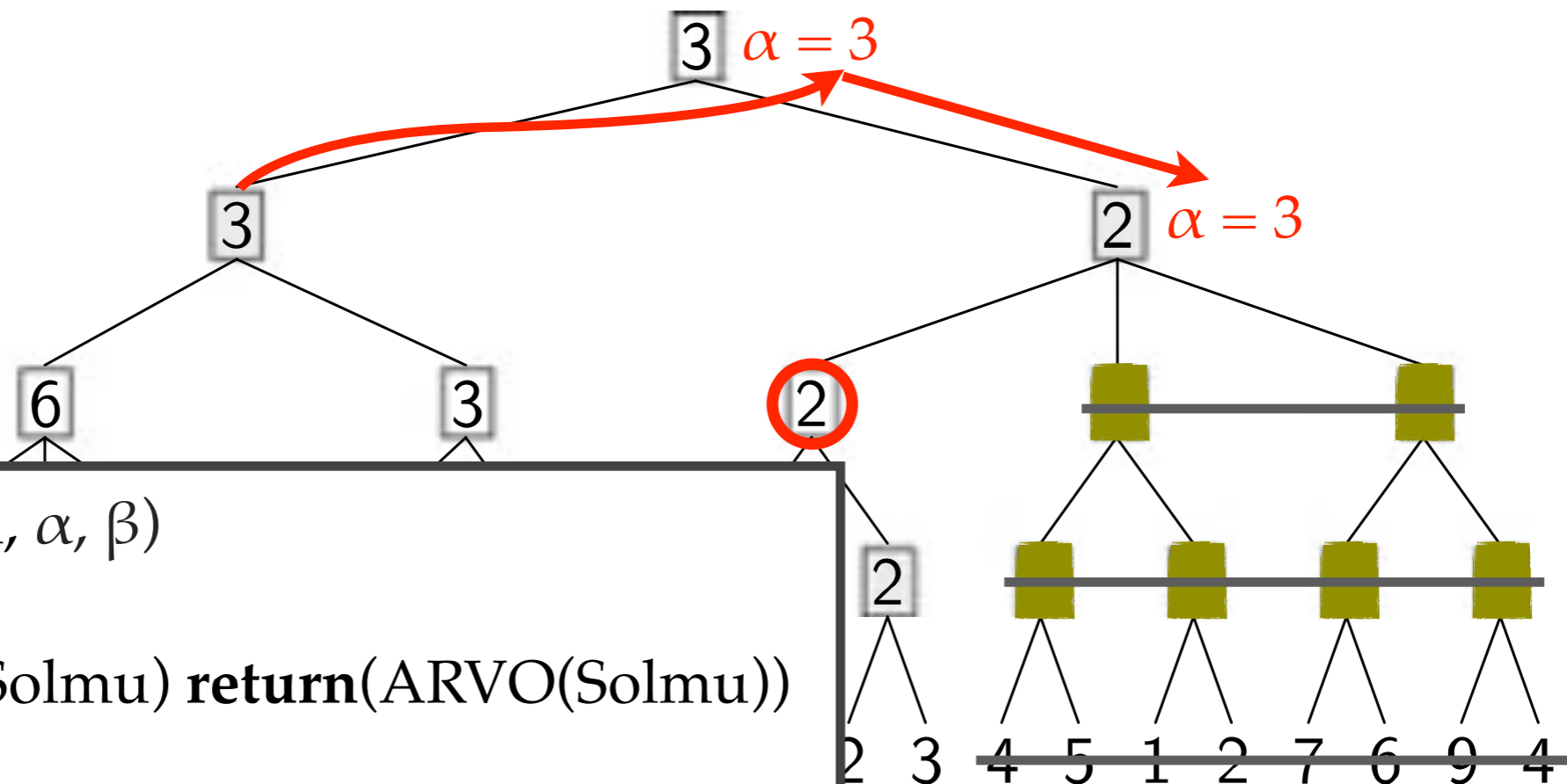
return(v)

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

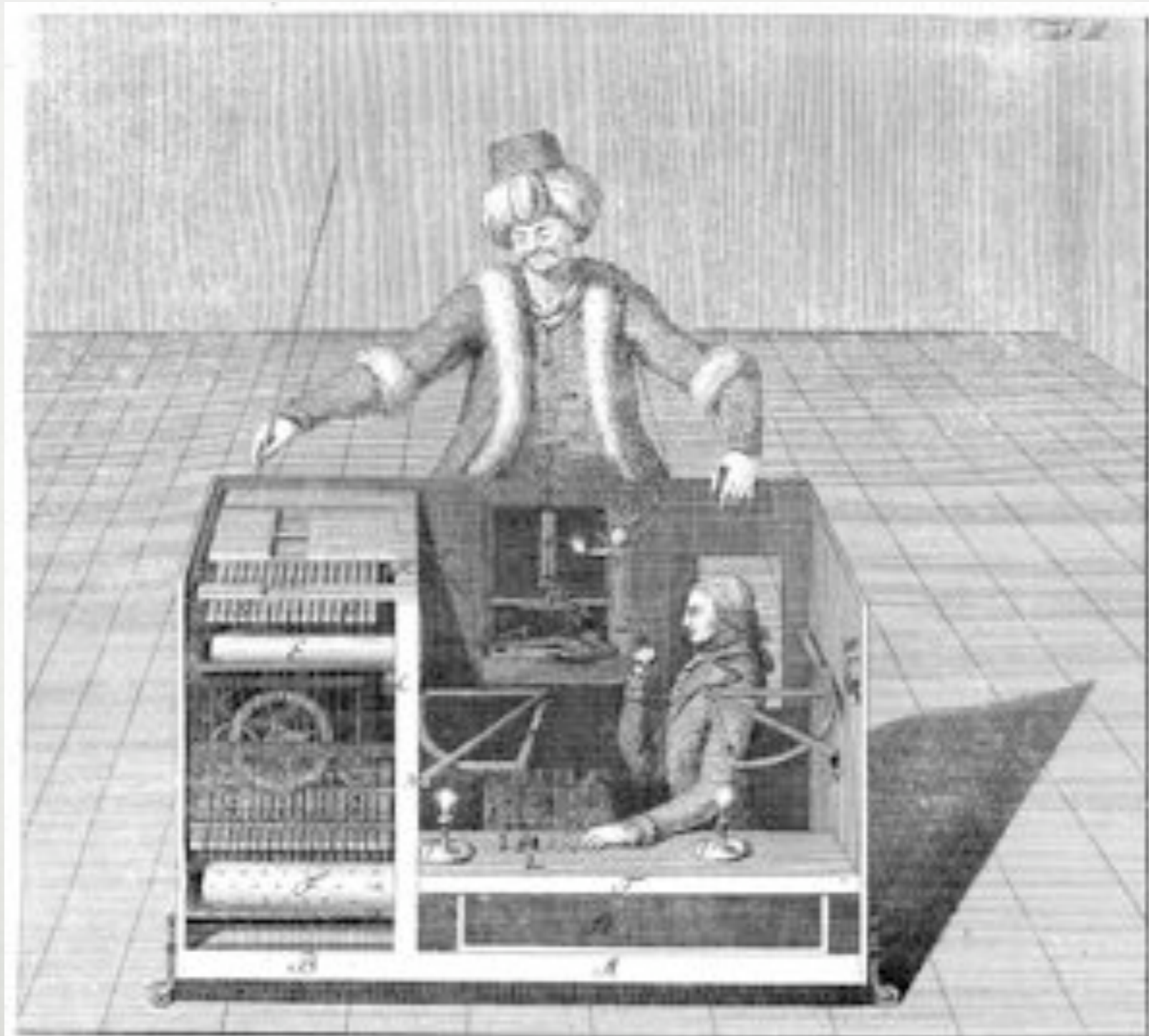
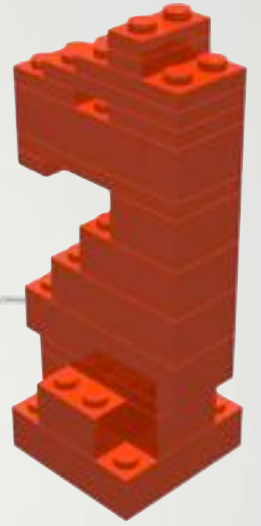
$\beta = \text{MIN}(\beta, v)$

return(v)

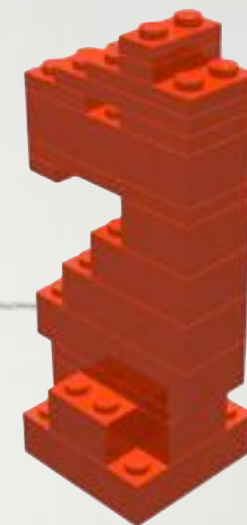
SHAKKI

The image is a screenshot of a YouTube video player. At the top left, the YouTube logo is visible. To its right is a search bar with the text 'Search' and 'Browse' next to it. Below the search bar, the video title 'Deep Blue beat G. Kasparov in 1997' is displayed in a large, bold font. Underneath the title, the channel name 'Eustake' is shown, followed by '6 videos' and a 'Subscribe' button. The main area of the player is a large black rectangle 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', and other standard video controls like '360p' and a full-screen icon. On the right side of the player, there is a 'Suggest' section with several video thumbnails, though they are mostly blank.

SHAKKI

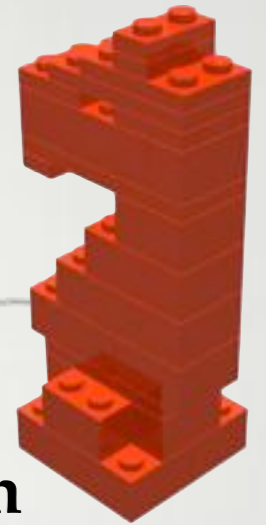


SHAKKI



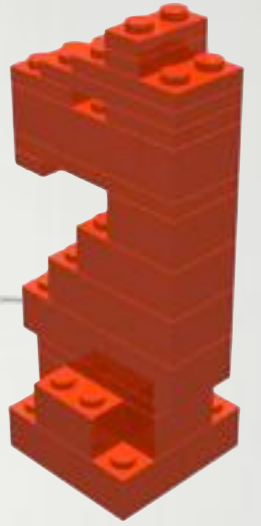
- 1769** **Wolfgang von Kempelen** rakentaa "Turkin"
- 1912** **L. Torres y Quevedo** rakentaa koneen kuningas&torni vs kuningas -loppupeleihin
- 1948** **Norbert Wiener** esittää syvyysrajoitetun minimax-algoritmin heuristisella arviontifunktiolla
- 1950** **Claude Shannon** julkaisee artikkelin "Programming 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



- 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

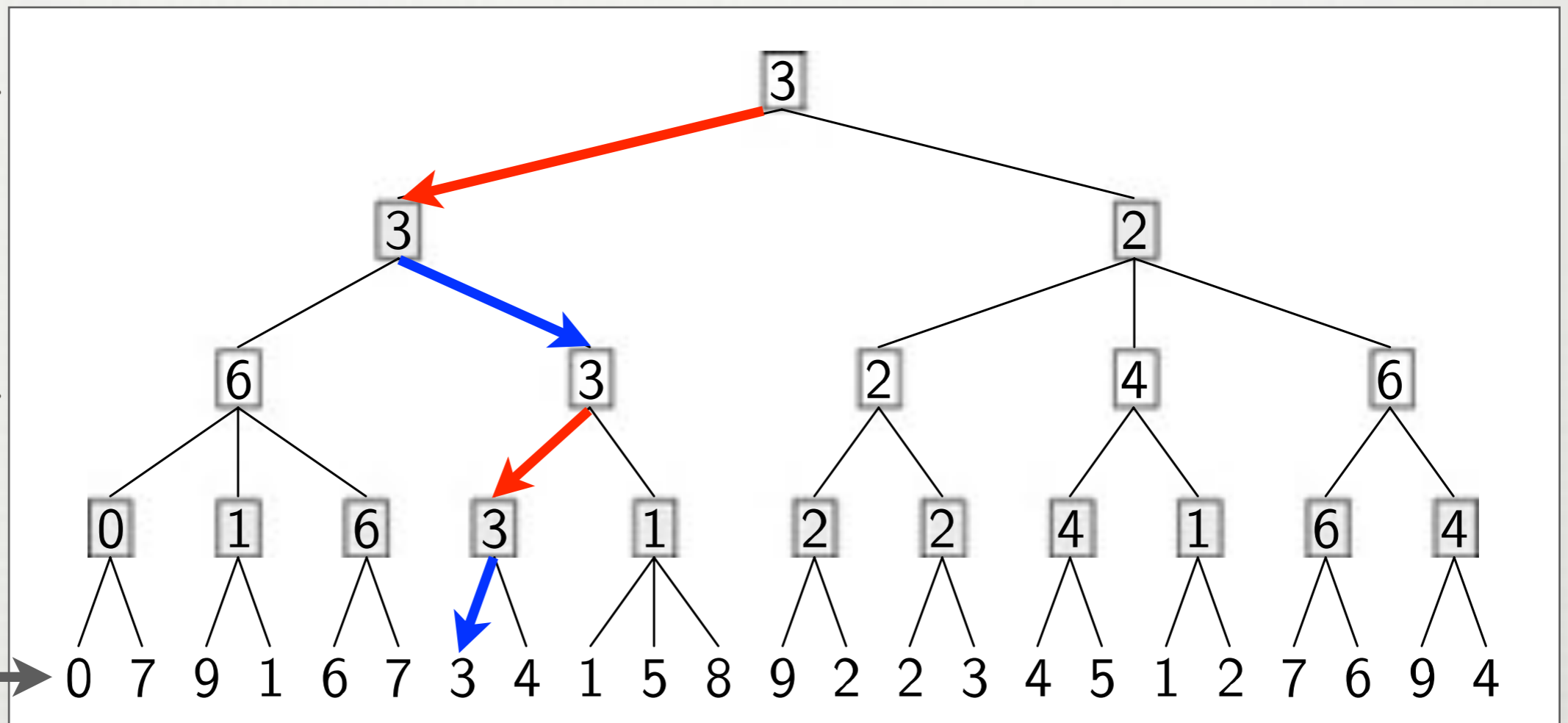
PELIPUU

MAX

MIN

MAX

MIN



ARVIOITA TILANTEEN HYVYYDESTÄ

ARPAPELIT

