

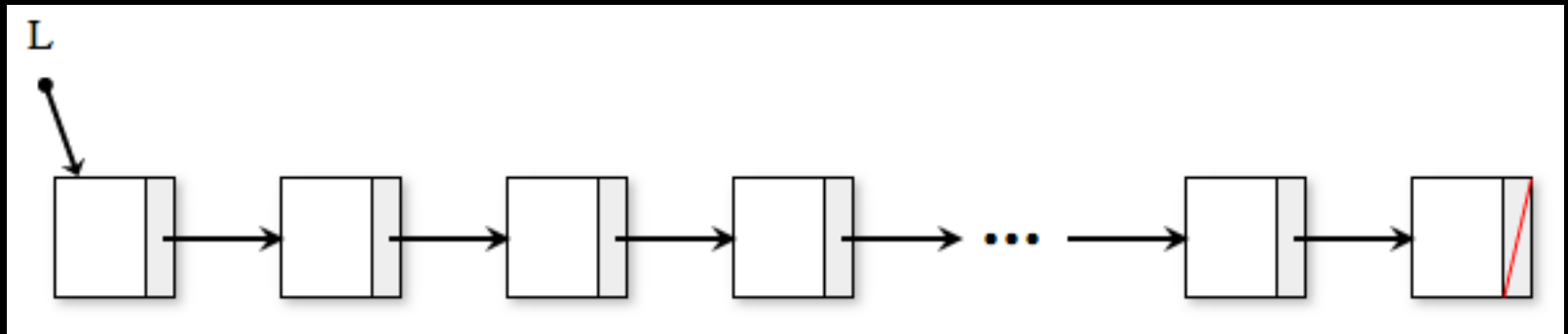
- Samanlainen kuin ViLLE-kotitehtäväkierros (pl. pisteet)
 - Sivuaine- ja pääaineopiskelijoille omat tenttikerrat
 - Luentosali (XXII tai pub1) ja K126C
 - Sivuaine: 21.10 klo 16:00-19:00
 - Pääaine: 24.10 klo 14:00-17:00
 - Ilmoittautuminen ViLLEssä
 - Sallitut materiaalit:
 - Pythonin dokumentaatio
 - Nelilaskin (Esim ViLLE:stä löytyvä) Ei graafinen tai kännykkä
- # TENTTI

PUUT JA REKURSIO

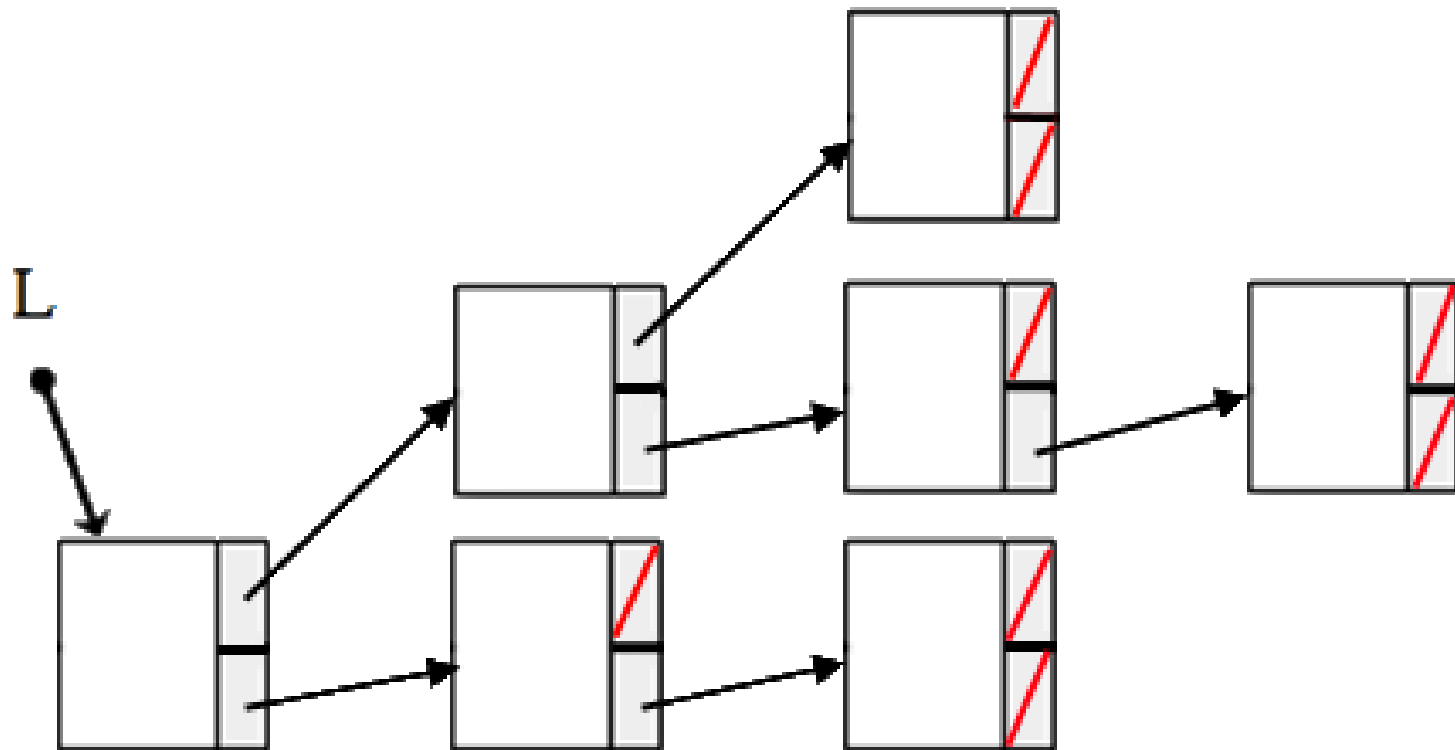
Myös kertaus

- Pikakertaus
- Puu
- Rekursio
- Kertaus

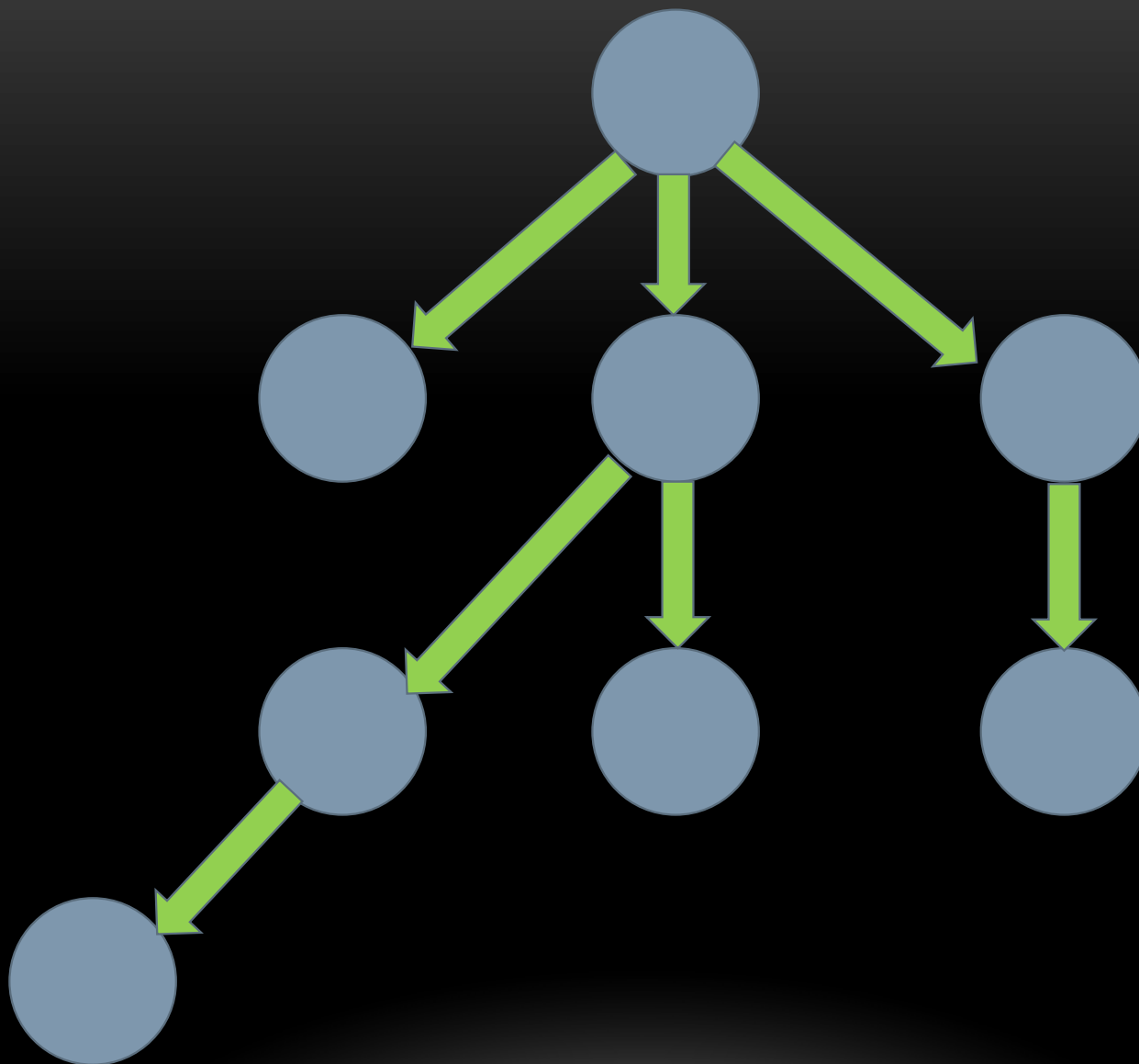
AGENDA



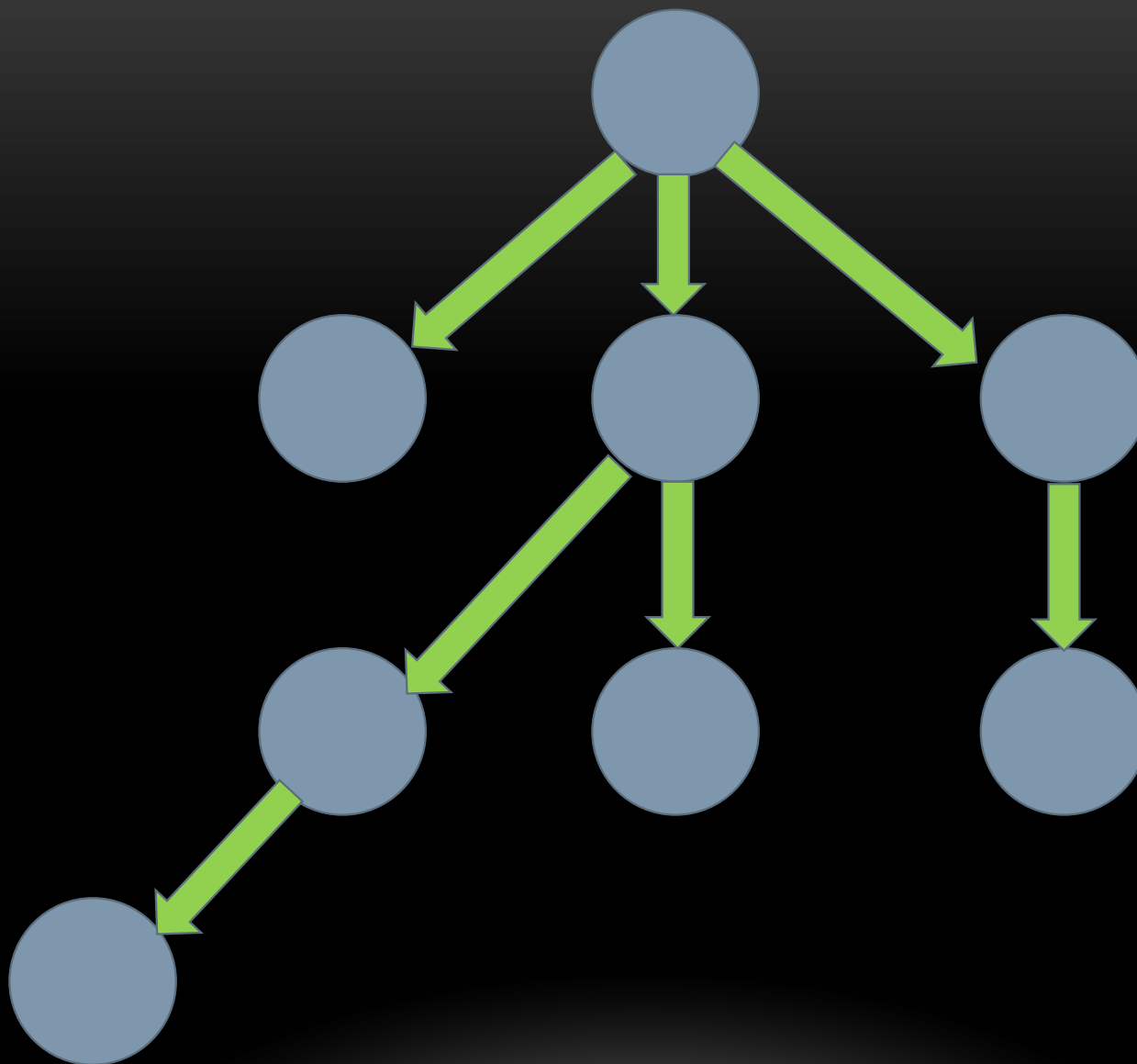
KERTAUS: LINKITETTY RAKENNE



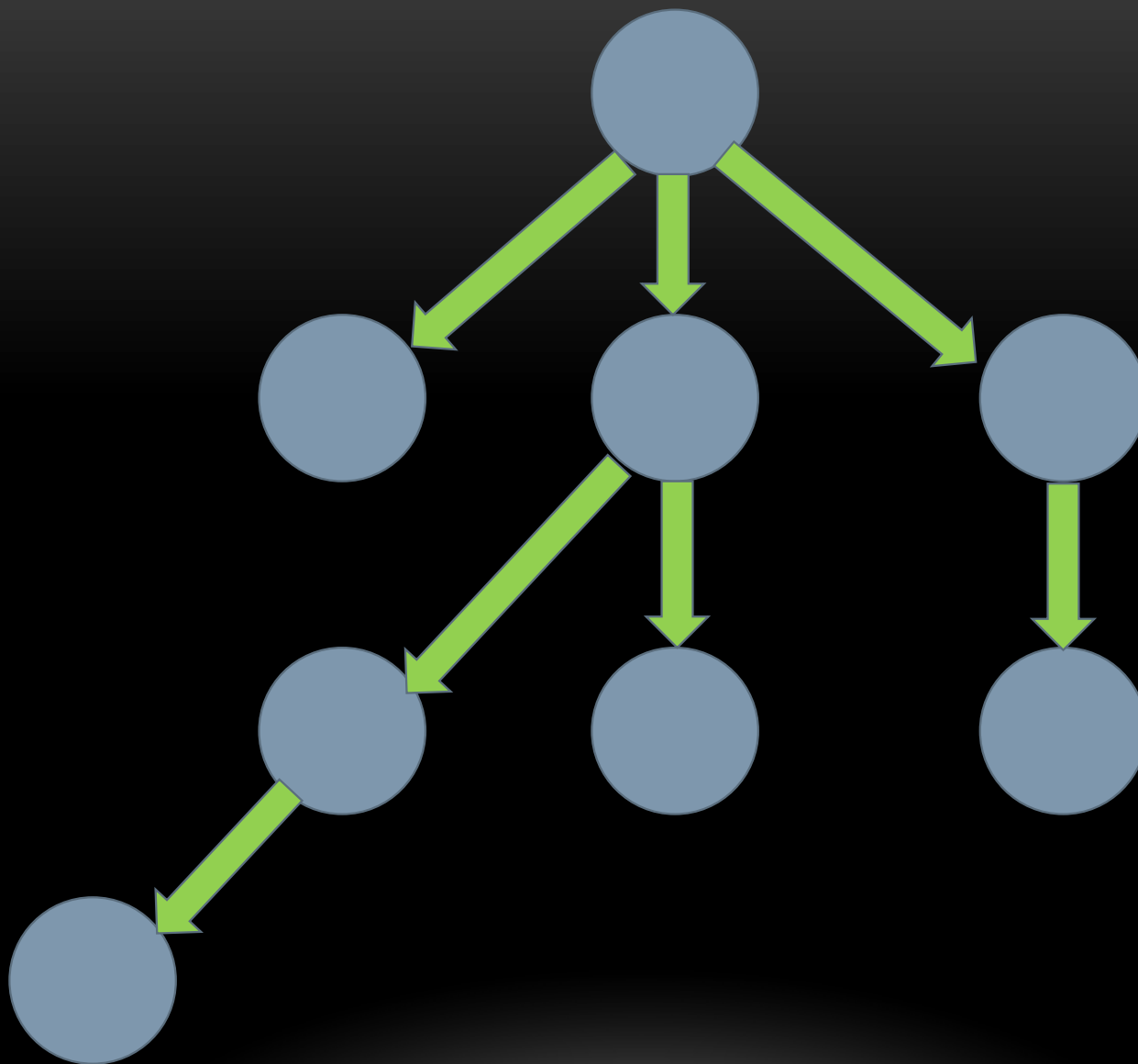
YLEINEN LINKITETTY RAKENNE



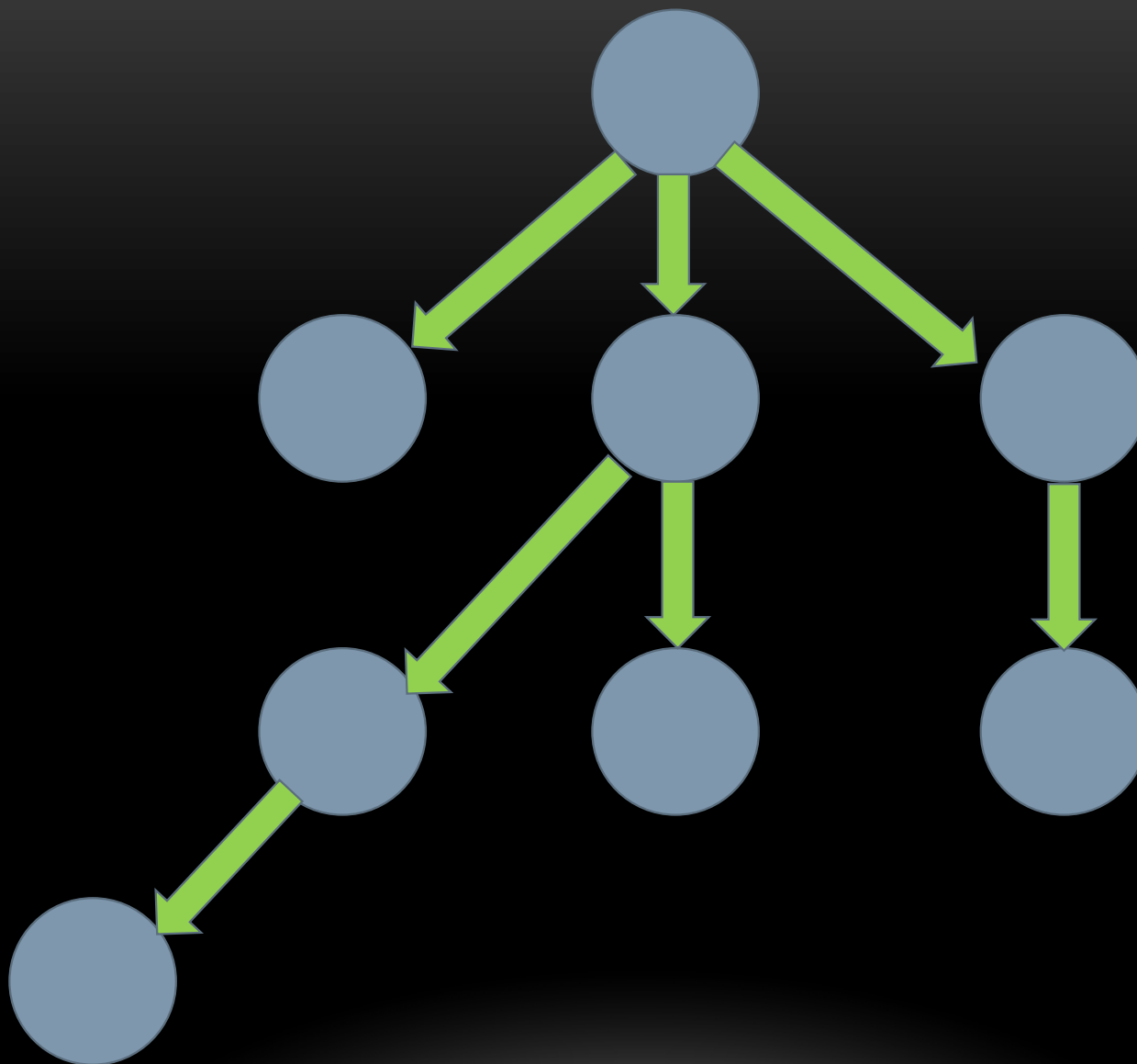
PUU



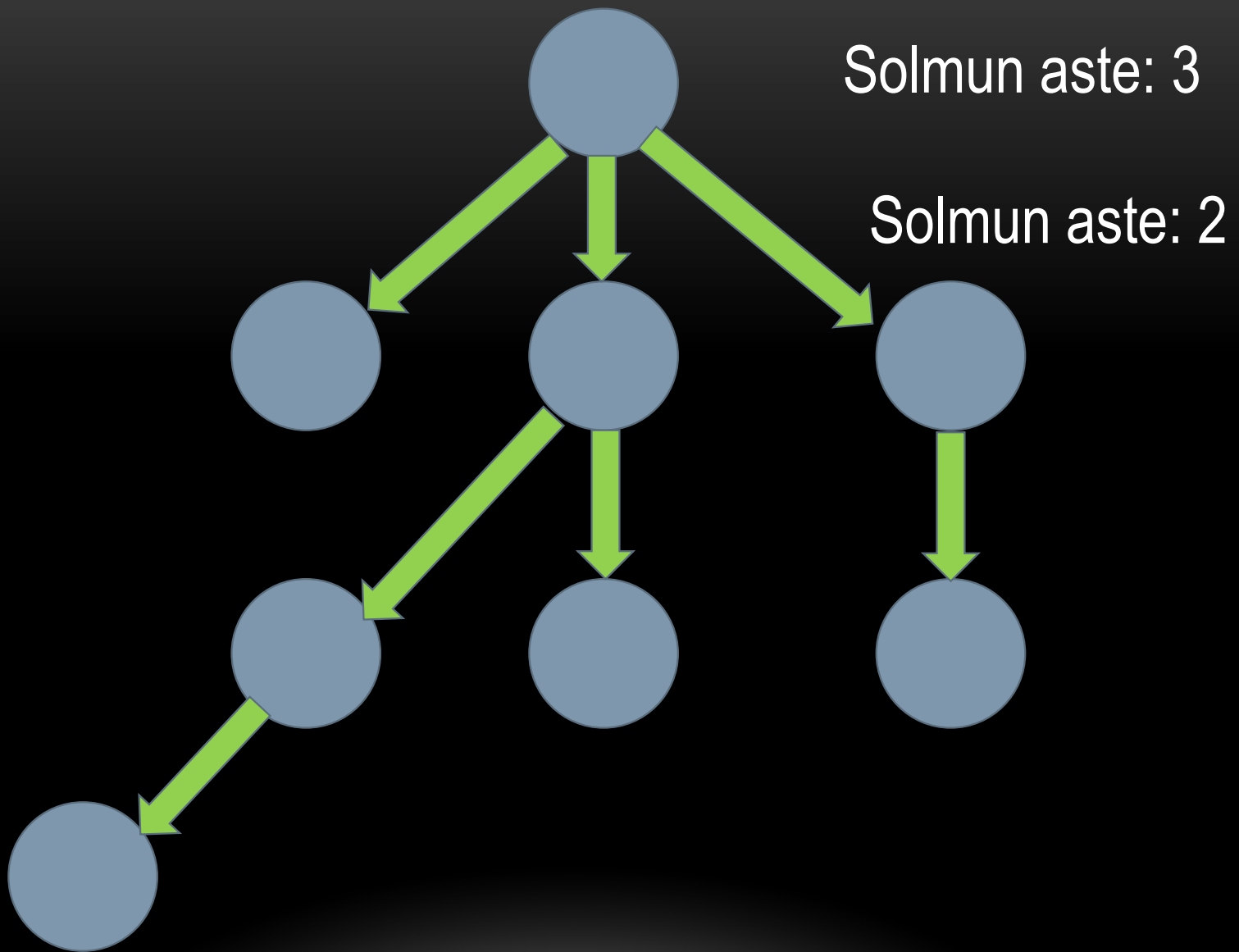
PUU; SOLMU (NODE)



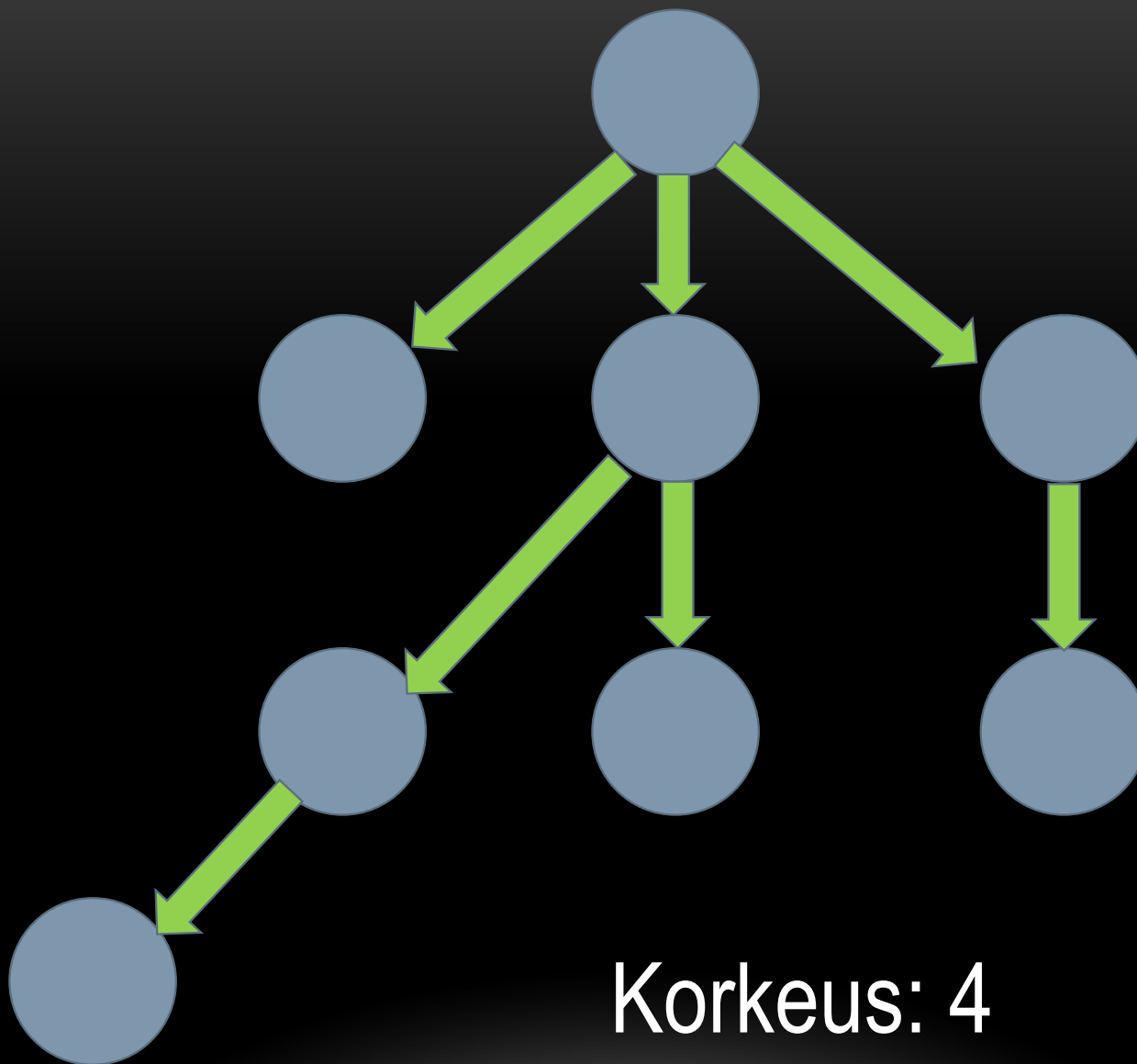
PUU; JUURISOLMU (ROOT)



PUU; LEHTISOLMU (LEAF)



PUU; SOLMUN ASTE (DEGREE)

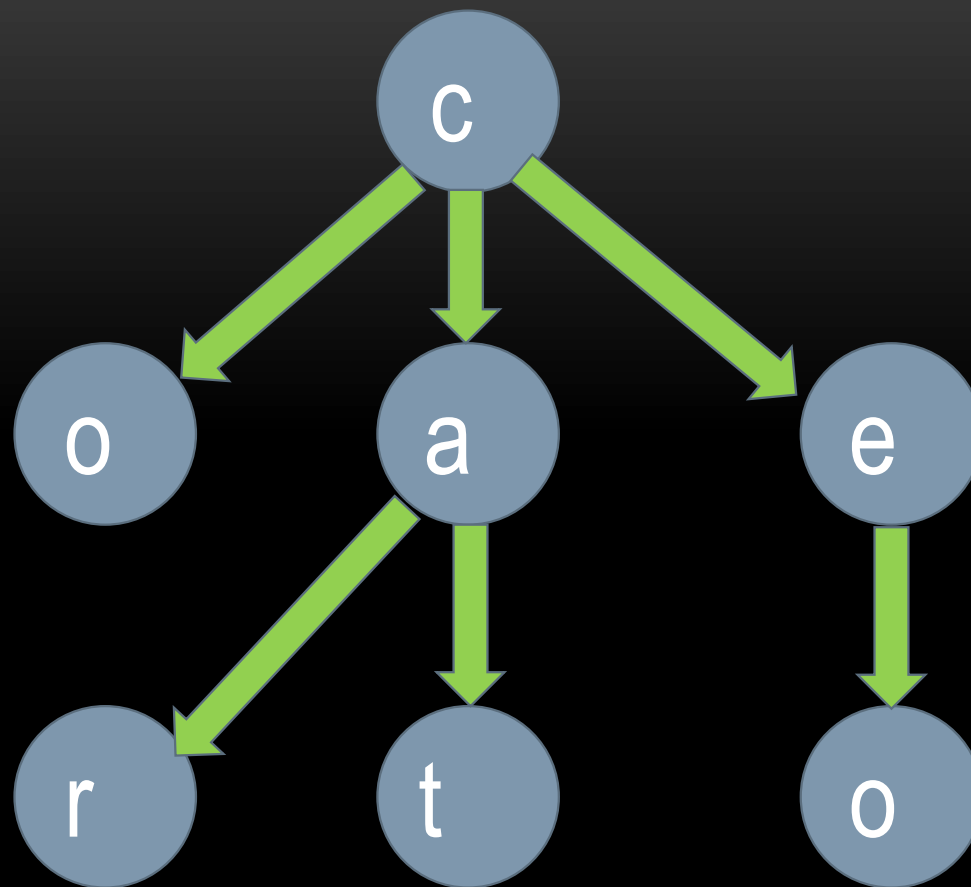


Korkeus: 4

PUU; KORKEUS (HEIGHT)

- Vain yksi juurisolmu
- Solmulla MAX yksi vanhempi
- Ei syklejä

PUUN MÄÄRITELMÄ



PUU



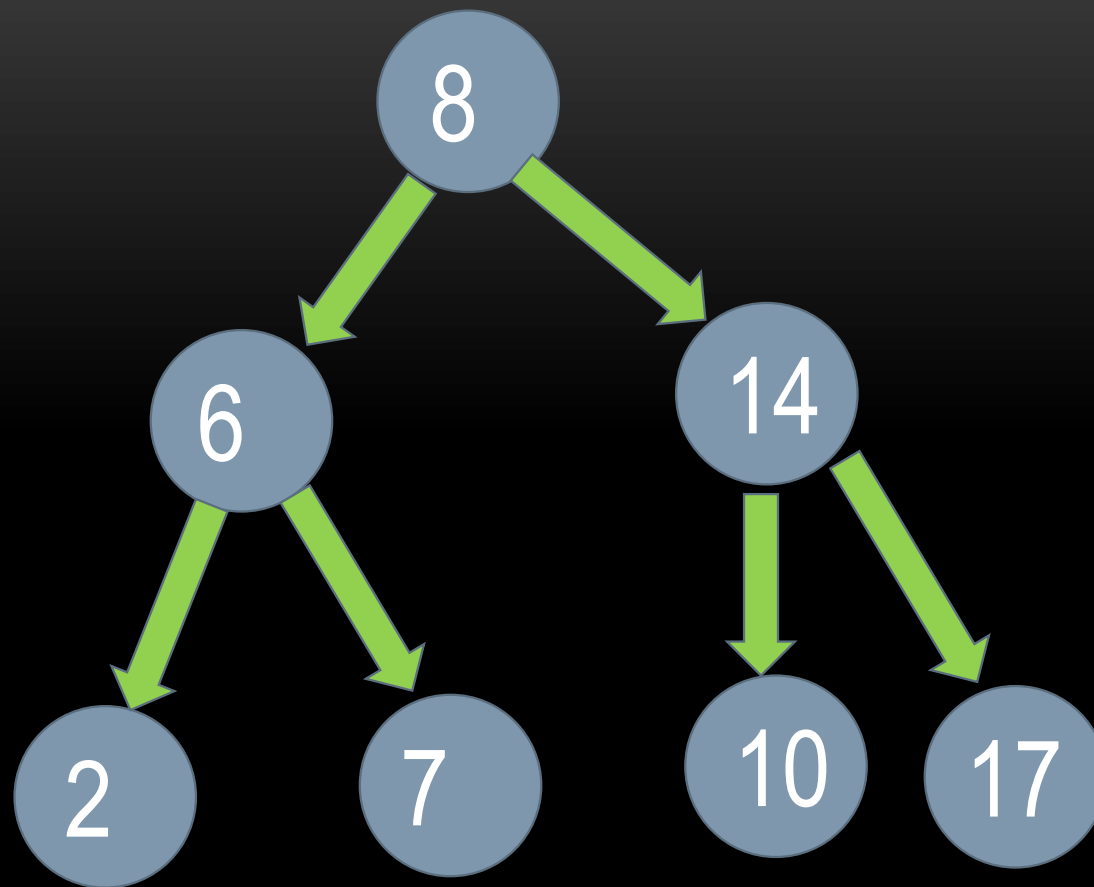
BINÄÄRIPUU

Puu, jonka aste on 2

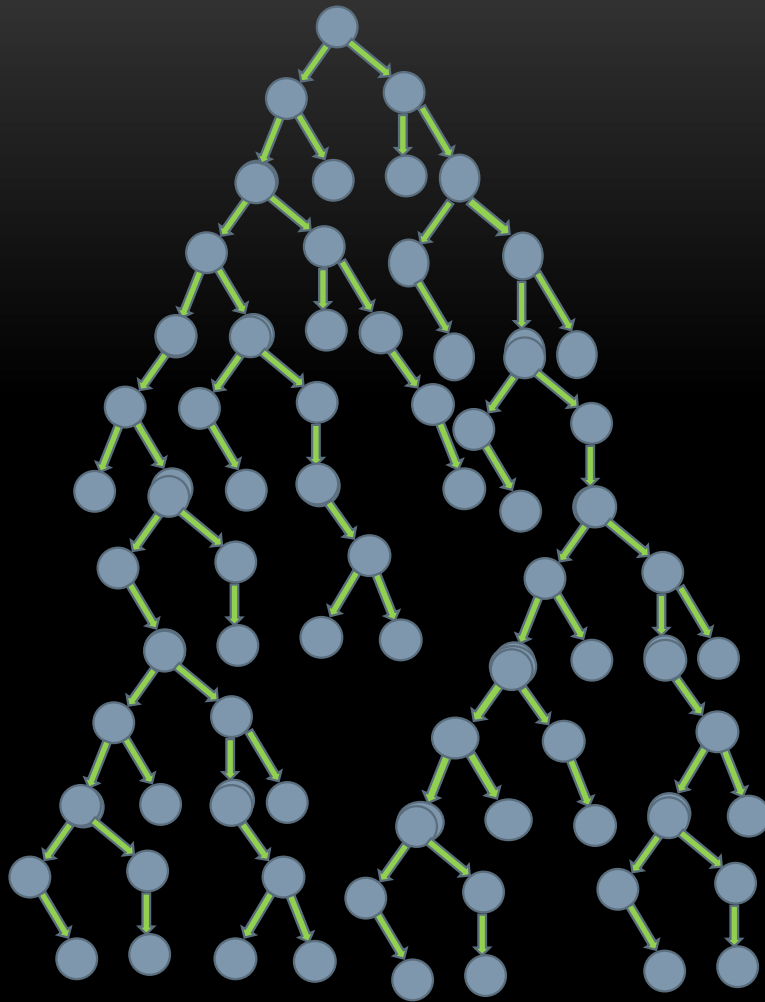
BINÄÄRIPUU

- Pienemmät arvot
vasempaan lapseen
- Suuremmat arvot
oikeaan lapseen

JÄRJESTETTY BINÄÄRIPUU



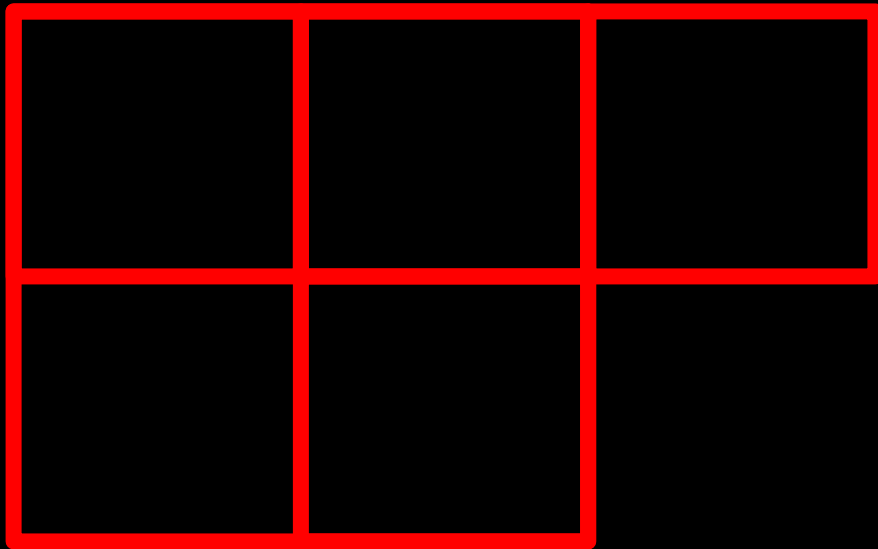
JÄRJESTETTY BINÄÄRIPUU



35 40
64
55

JÄRJESTETTY BINÄÄRIPUU

Kuinka monta neliötä?

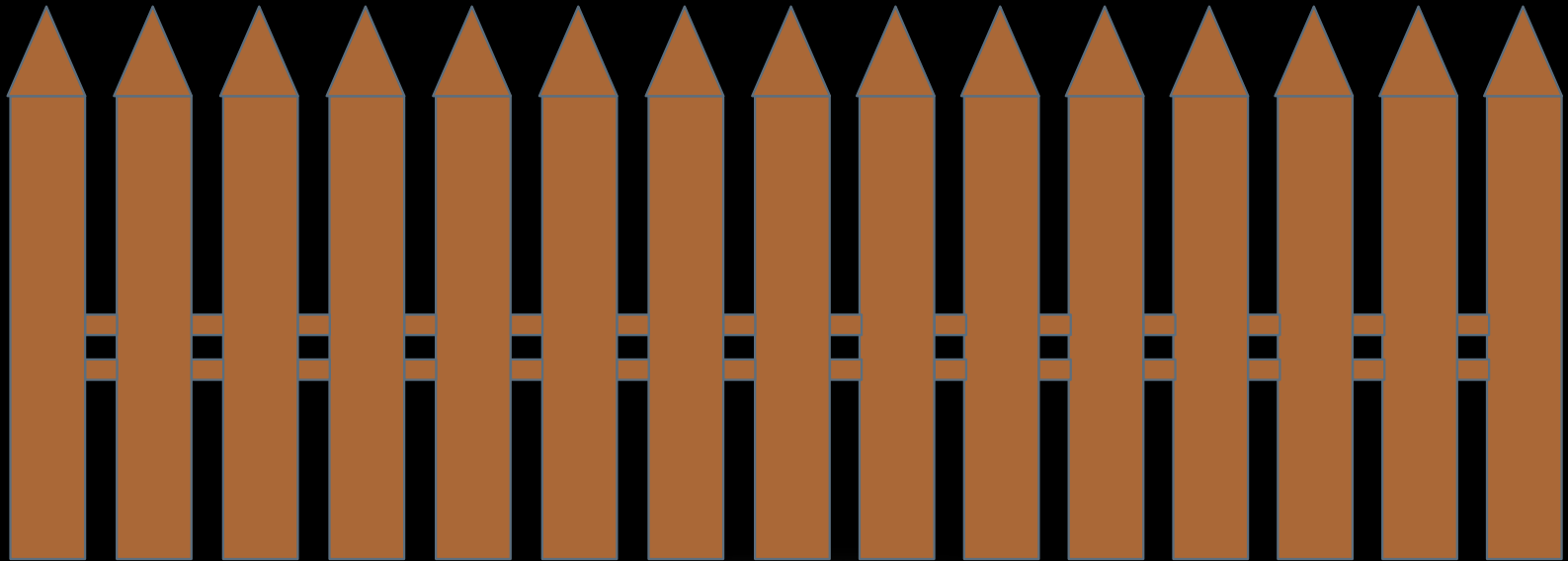


6

TAJUNTA-HARJOITE

Aidan maalaaminen?

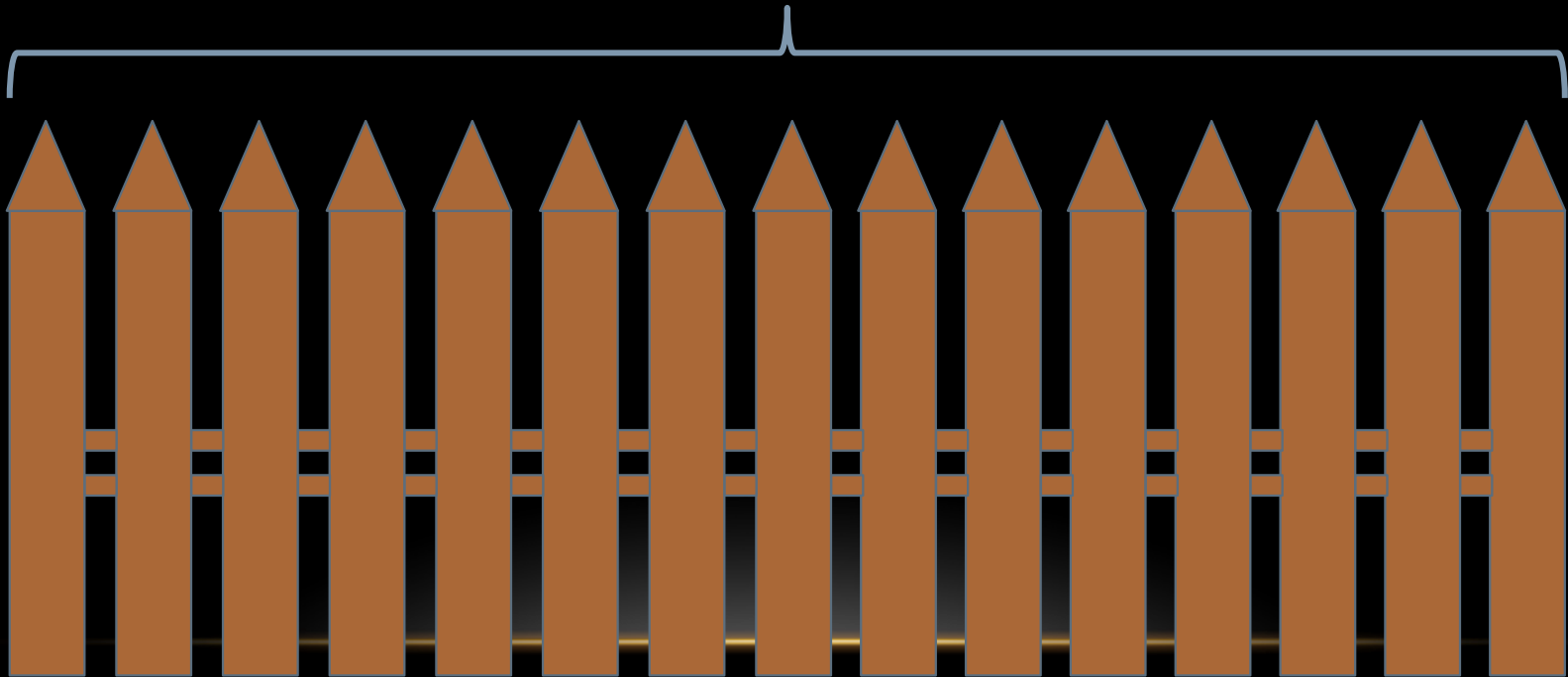
Ratkaisu: Aidan maalaamattomien
lautojen maalaaminen



REKURSIO

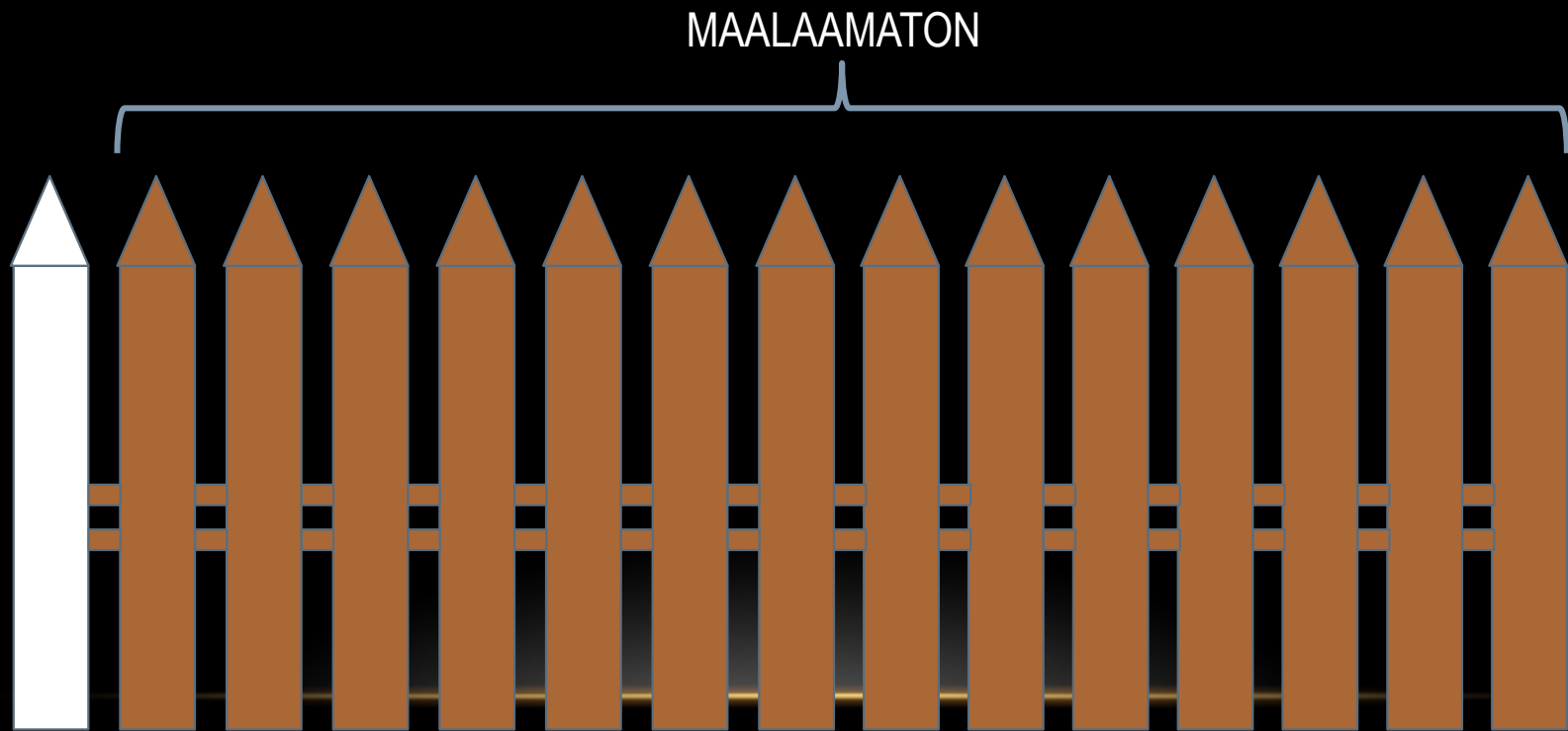
```
def maalaaAita(maalaamaton):  
    if(maalaamaton not empty):  
        maalaa1Lauta(maalaamaton)  
        maalaaAita(maalaamaton)
```

MAALAAMATON



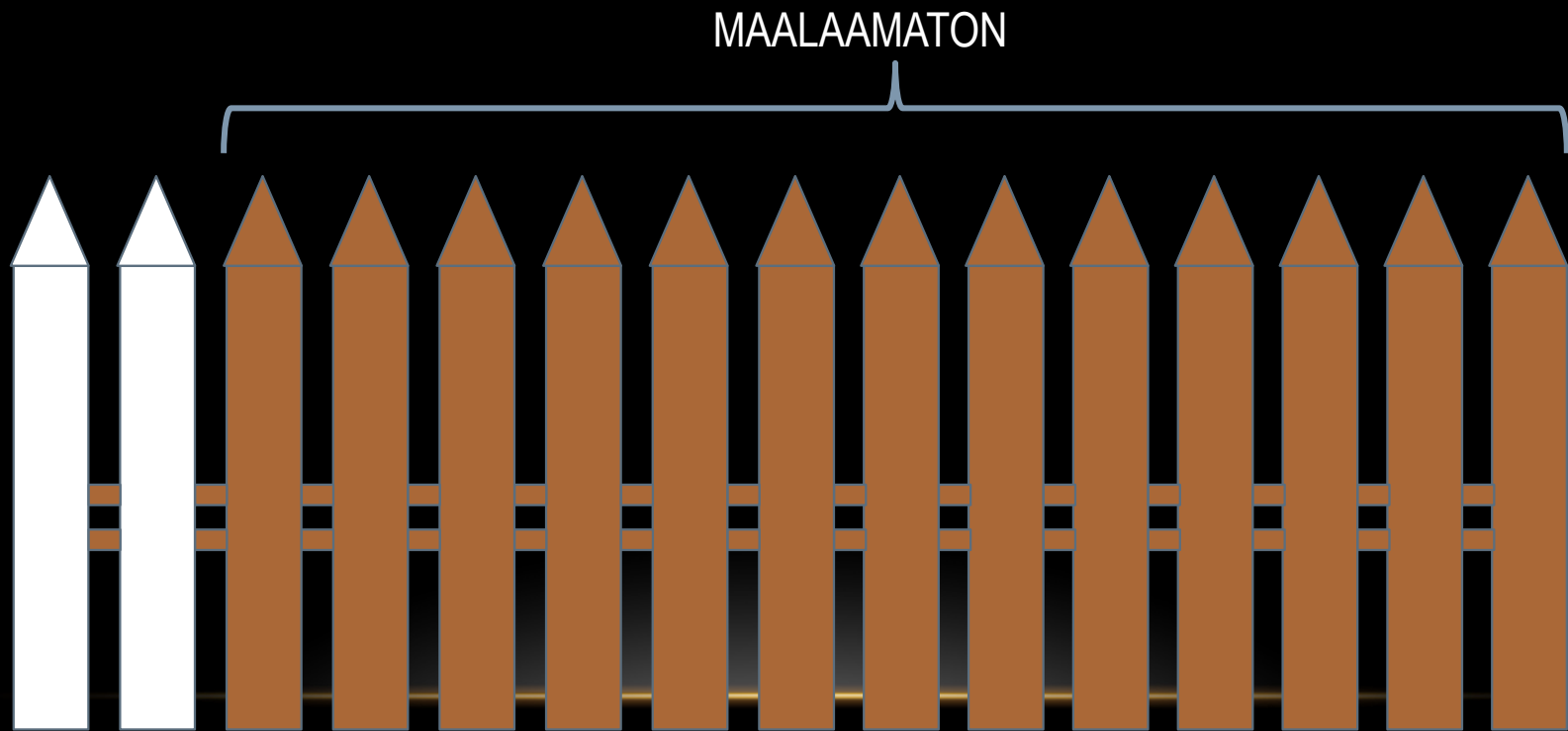
REKURSIO

```
def maalaaAita(maalaamaton):  
    if(maalaamaton not empty):  
        maalaa1Lauta(maalaamaton)  
        maalaaAita(maalaamaton)
```



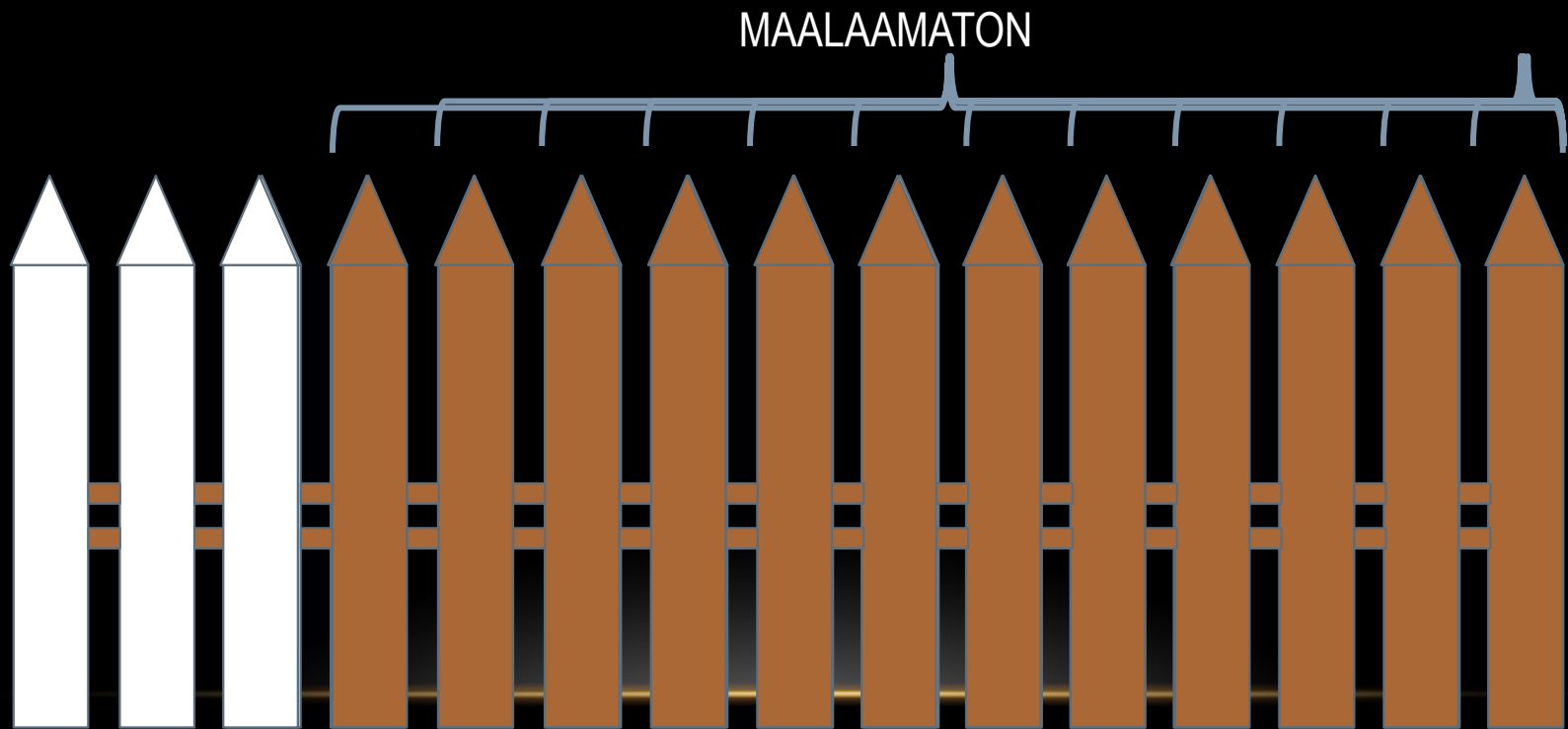
REKURSIO

```
def maalaaAita(maalaamaton):  
    if(maalaamaton not empty):  
        maalaa1Lauta(maalaamaton)  
        maalaaAita(maalaamaton)
```



REKURSIO

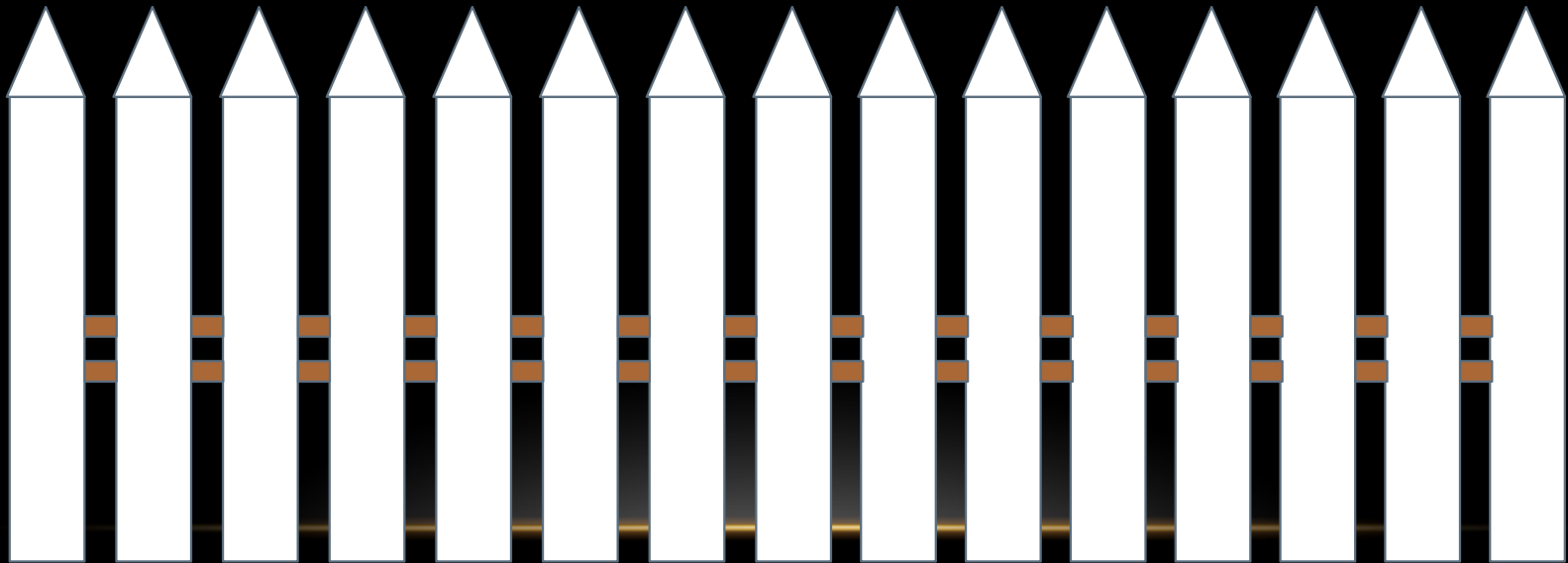
```
def maalaaAita(maalaamaton):  
    if(maalaamaton not empty):  
        maalaa1Lauta(maalaamaton)  
        maalaaAita(maalaamaton)
```



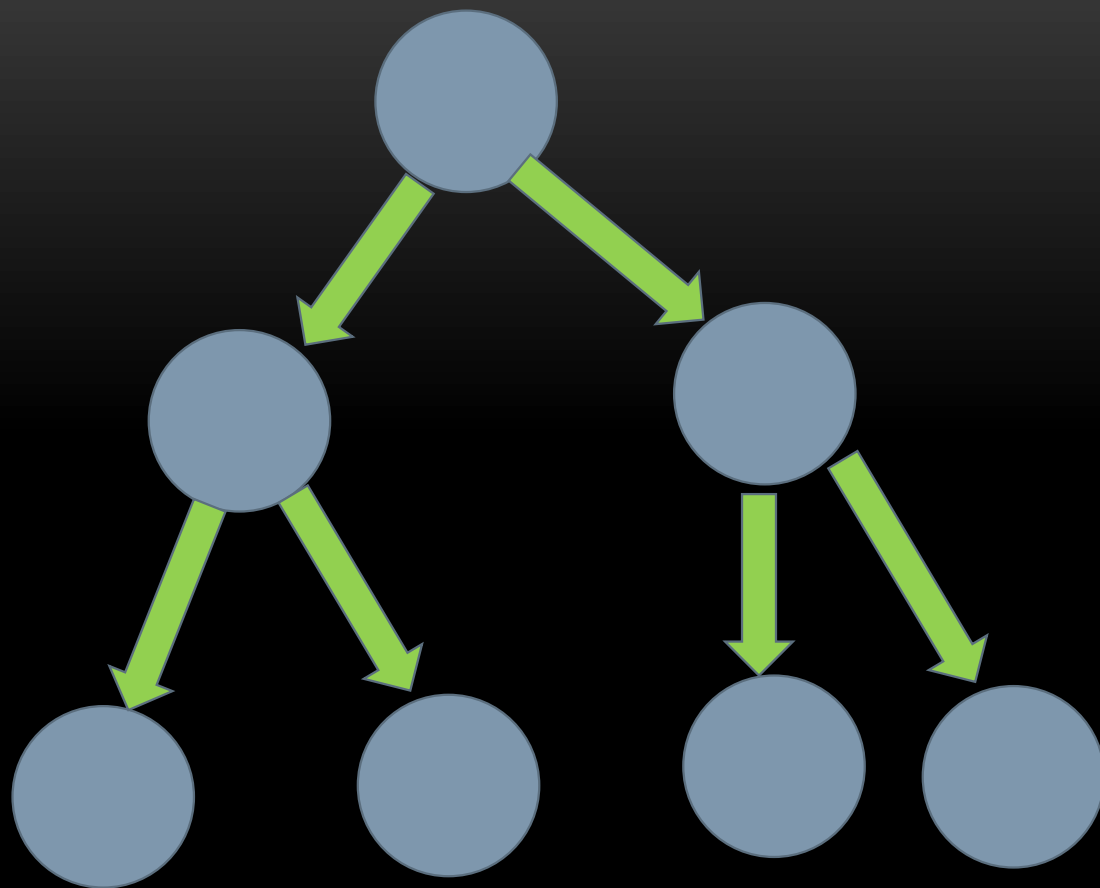
REKURSIO

```
def maalaaAita(maalaamaton):  
    if(maalaamaton not empty):  
        maalaa1Lauta(maalaamaton)  
        maalaaAita(maalaamaton)
```

MAALAAMATON



REKURSIO



REKURSIO

77?

51

REKURSIO

```
def etsiPuusta(alkio):  
    if alkio == 77  
        return True  
    return False
```

51

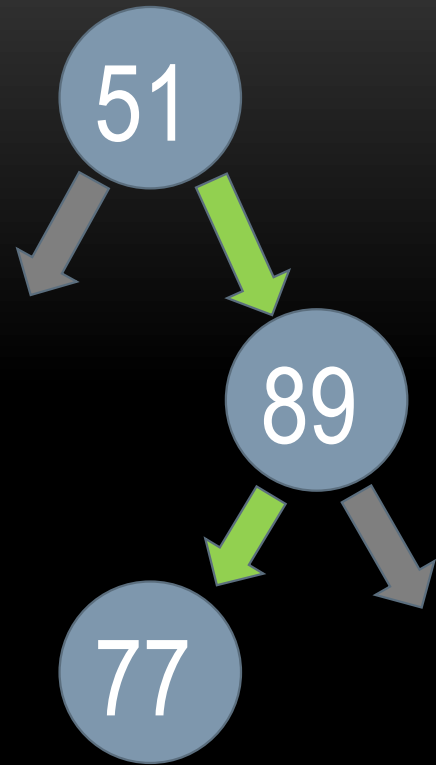
REKURSIO

```
def etsiPuusta(alkio):  
    if alkio == 77:  
        return True  
    if alkio > 77:  
        return etsiPuusta(alkio.vasen)  
    if alkio < 77:  
        return etsiPuusta(alkio.oikea)  
    return False
```



REKURSIO

```
def etsiPuusta(alkio):  
    if alkio == 77:  
        return True  
    if alkio > 77:  
        return etsiPuusta(alkio.vasen)  
    if alkio < 77:  
        return etsiPuusta(alkio.oikea)  
    return False
```



REKURSIO

0	1	2
6	54	-2

Rekursiivinen metodi kokonaislukulistan
summan laskemiseksi?

REKURSIO

0

1

2

6

54

-2

```
def summaaTaulukko(taulu, indeksi):  
    if indeksi <= len(taulu)-1:  
        print "Summataa",taulu[indeksi]  
        summa = summaaTaulukko(taulu, indeksi+1)  
        print taulu[indeksi], "Summattu"  
        return summa + taulu[indeksi]  
    return 0
```

REKURSIO

```
print summaTaulukko([1,2], 0):
```

```
1 def summaaTaulukko(taulu, indeksi): #indeksi == 0
2     if indeksi <= len(taulu)-1:
3         print "Summataa",taulu[indeksi]
4         summa = summaaTaulukko(taulu, indeksi+1)
14        print taulu[indeksi], "Summattu"
15        return summa + taulu[indeksi]
        return 0
```

```
5 def summaaTaulukko(taulu, indeksi): #indeksi == 1
6     if indeksi <= len(taulu)-1:
7         print " Summataa",taulu[indeksi]
8         summa = summaaTaulukko(taulu, indeksi+1)
12        print taulu[indeksi], " Summattu"
13        return summa + taulu[indeksi]
        return 0
```

```
9 def summaaTaulukko(taulu, indeksi): #indeksi == 2
10     if indeksi <= len(taulu)-1: #FALSE
        print " Summataa",taulu[indeksi]
        summa = summaaTaulukko(taulu, indeksi+1)
        print taulu[indeksi], " Summattu"
        return summa + taulu[indeksi]
11 return 0
```

REKURSION SUORITUS

Kurssilla käydyt asiat

- Kontrollirakenteet
- Modulaarisuus
- Binääriesitykset
- Tietorakenteet

KERTAUS

Tallennuspaikka
muistissa,
jolle annettu nimi.

MUUTTUJA

FOR syntaksi

FOR silmukkamuuttuja IN lista:

lohko

SILMUKAT

WHILE syntaksi

Two vertical blue bars of different heights are positioned to the left of the text.

WHILE ehto:

lohko

SILMUKAT

Proseduuri

- Ei palauta arvoa
- Lause

Funktio

- Palauttaa arvon
- Lauseke

METODIT

Syntaksi

DEF metodinNimi(parametrit):
 lohko

METHOD

Tallennusrakenteet:

- Muistipaikka (perustyyppi)
- Taulukko (viittaus)
- Linkitetty rakenne (viittaus)

TIETORAKENTEET

Abstraktit tietorakenteet

- Lista, Jono, Pino
- Kuvaus
- Puu
- ...

TIETORAKENTEET

Tallennustapa ei kiinnosta*;

Tietorakenteen operaatiot (ja niiden vaikutus) kiinnostaa*

*eli määritelty

TIETORAKENTEET

Koodi (#01- #05):
lista = [4,3,9]

Muuttujat:
lista->#06->link

#01	#02	#03	#04	#05
1100	1011	1101	1000	0000
#06	#07	#08	#09	#0A
0100	1100	0101	1001	1010
#0B	#0C	#0D	#0E	#0F
1001	0011	1111	1001	1001
#10	#11	#12	#13	#14
0000	0100	0011	1001	0000

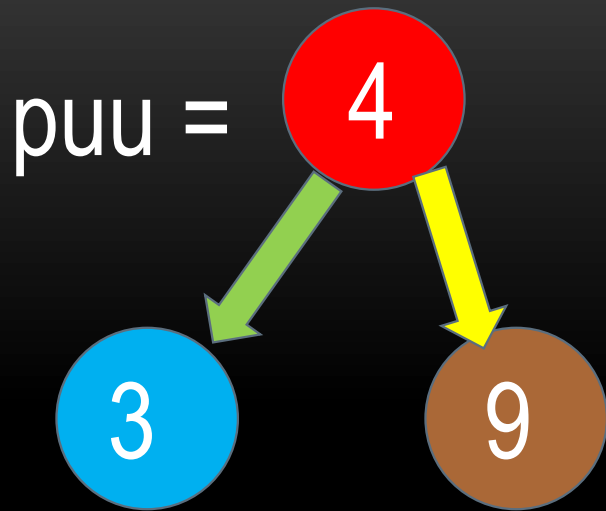
LINKITETTY RAKENNE

lista = [4,3,9]

Muuttujat:
lista->#06->link

#01	#02	#03	#04	#05
1100	1011	1101	1000	0000
#06	#07	#08	#09	#0A
0100	1100	0101	1001	1010
#0B	#0C	#0D	#0E	#0F
1001	0011	1111	1001	1001
#10	#11	#12	#13	#14
0000	0100	0011	1001	0000

LINKITETTY RAKENNE



Muuttujat:
puu->#06->2link

#01	#02	#03	#04	#05
1100	1011	1101	1000	0000
#06	#07	#08	#09	#0A
0100	1100	1111	1001	1010
#0B	#0C	#0D	#0E	#0F
1001	0011	0000	0000	1001
#10	#11	#12	#13	#14
0000	0000	0011	1001	0000

LINKITETTY RAKENNE

	Muistip aikka	Taulukko	Linkitetty rakenne
Laa- jen- tuva	ei	-	+
Haku	+	+	-
Tila- tarve	+	+	-/+