

Rekursio (perusteiden kertaus)

COMP.CS.300 Tietorakenteet ja algoritmit 1

Matti Rintala (matti.rintala@tuni.fi)

Mitä on rekursio?

- "Funktio, joka kutsuu itseään"?

```
int fibonacci(int n)
{
    if (n<2)
    { return 1; }
    else
    {
        return fibonacci(n-1) +
            fibonacci(n-2);
    }
}
```

Mitä on rekursio?

- Asia, jonka *osat* ovat **samanlaisia** kuin *kokonaisuus* (vain pienempiä)

Rekursiiviset funktiot

- Useita funktio suorituksia kesken "sisäkkäin"
- Joka kutsulla omat parametrit, omat paikalliset muuttujat
- "Kutsupino"
- Triviaalitapaus!

Rekursiiviset funktiot

- Rekursio \rightarrow silmukka? ("häntärekursio")
- Silmukka \rightarrow rekursio?

Puolitushaku

BinarySearch(*A*, *left*, *right*, *value*)

(*A* nousevassa järjestyksessä)

```

1 if left = right then                (vain yksi alkio)
2     if A[left] = value then        (ainoa alkio on etsitty)
3         return left                  (palauta paikka)
4     else
5         ▷ Ei löytynyt!
6 else
7     middle :=  $\lfloor (\textit{left} + \textit{right}) / 2 \rfloor$     (alue puoliksi)
8     if value ≤ A[middle] then
9         return BinarySearch(A, left, middle, value)
10    else
11    return BinarySearch(A, middle+1, right, value)

```

BinarySearch(*A*, *left*, *right*, *value*)

(*A* nousevassa järjestyksessä)

```
1 if left = right then  
2     if A[left] = value then  
3         return left  
4     else  
5         ▷ Ei löytynyt!  
6 else  
7     middle :=  $\lfloor (left+right)/2 \rfloor$   
8     if value ≤ A[middle] then  
9         return BinarySearch(A, left, middle, value)  
10    else  
11    return BinarySearch(A, middle+1, right, value)
```


2	4	5	6	8	9	9	10	15
---	---	---	---	---	---	---	----	----

Hajoita ja hallitse -suunnitteluperiaate

COMP.CS.300 Tietorakenteet ja algoritmit 1

Matti Rintala (matti.rintala@tuni.fi)

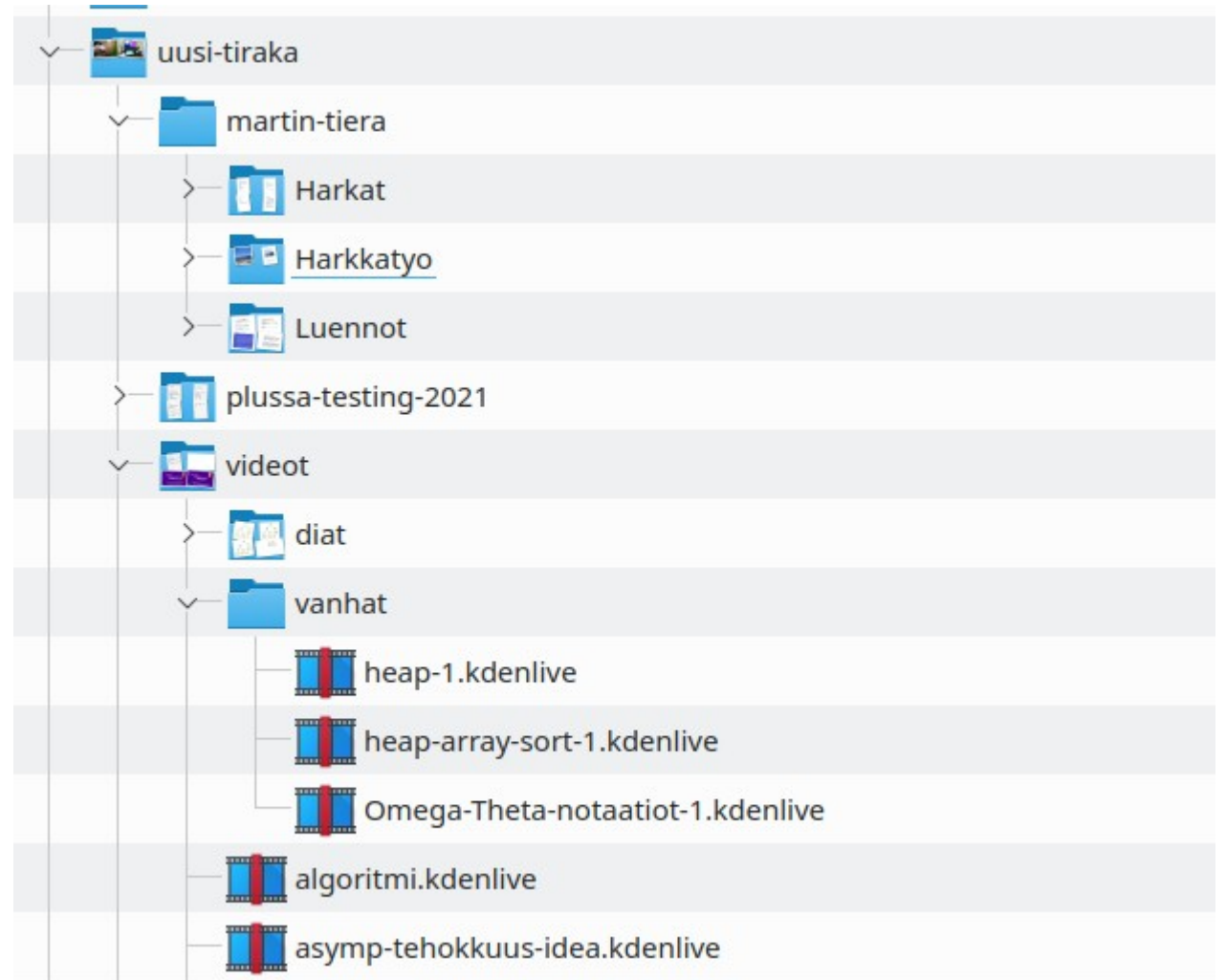
Hajoita ja hallitse

- Jako (noin) *yhtäsuuriin samanlaisiin* osaongelmiin
- Pienimmät ongelmat triviaaleja
- Isommat ratkaistaan rekursiivisesti
- Lopuksi kootaan ratkaisu osaratkaisuista
- (*Kaikki* rekursio ei ole hajoita & hallitse)

***Miksi* hajota & hallitse (rekursio)?**

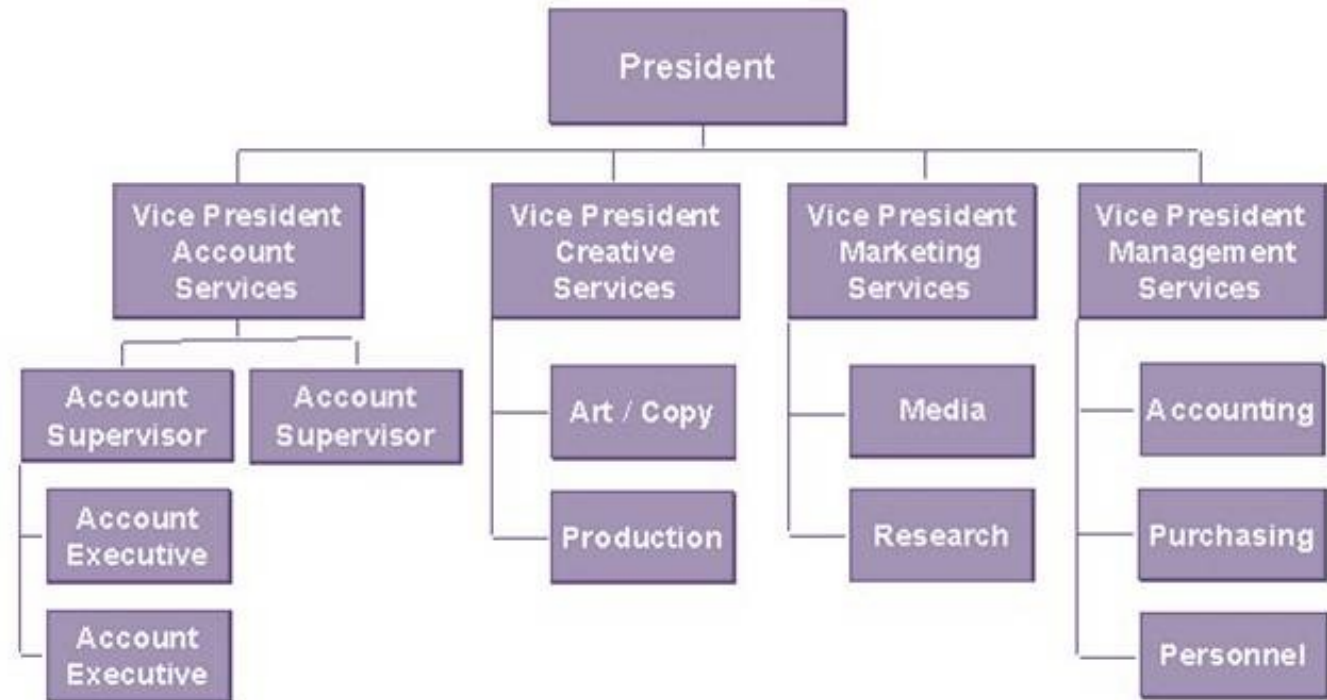
- *Rekursiiviset* ongelmat
- *Hierarkkiset* ongelmat

Miksi hajota & hallitse (rekursio)?



Miksi hajoitaa & hallitse (rekursio)?

Agency Department System



Miksi hajoitse & hallitse (rekursio)?



(Photo by Leon Brocard)

Miksi hajota & hallitse (rekursio)?

- Algoritmit, joiden **tehokkuus** perustuu hajota & hallitse -ideaan
- Yleensä rekursiivisia algoritmeja
- Esim. puolitushaku vs lineaarinen haku

2	4	5	6	8	9	9	10	15
---	---	---	---	---	---	---	----	----

Rekursiiviset tietorakenteet

- *Hierarkiset* tietorakenteet, esim. *puut*
- Hierarkisten tietorakenteiden algoritmit (usein) rekursiivisia

