

STL:n algoritmit

COMP.CS.300 Tietorakenteet ja algoritmit 1

Matti Rintala (matti.rintala@tuni.fi)

- **Paljon** valmiita algoritmeja
- *Tutustu* algoritmistaan
(esim. cppreference.com)
- Parametreina iteraattoreita
- Eivät koskaan lisää/poista alkioita!
(mitätöityminen)
- (Tarjolla myös rinnakkaisuutta
hyödyntäviä versioita)

STL:n algoritmit - esimerkkejä

COMP.CS.300 Tietorakenteet ja algoritmit 1

Matti Rintala (matti.rintala@tuni.fi)

Selection-Sort(A)

```
1 for next_elem := 1 to A.length-1 do
2   smallest := next_elem
3   for place := next_elem+1 to A.length do
4     if A[place] < A[smallest] then
5       smallest := place
6   A[next_elem]  $\leftrightarrow$  A[smallest]
```

(siirretään osien välistä rajaa)

(mahd. pienimmän alkion paikka = eka alkio)

(jos löytyy vielä pienempi alkio...)

(...otetaan sen paikka talteen)

(vaihdetaan pienin alkio alkuun)

Mergesort(*A*, *left*, *right*)

- 1 **if** *left* < *right* **then** (triviaalitapaukselle ei tehdä mitään)
- 2 *mid* := $\lfloor (\textit{left} + \textit{right}) / 2 \rfloor$ (lasketaan puoliväli)
- 3 Mergesort(*A*, *left*, *mid*) (järjestetään vasen puoli)
- 4 Mergesort(*A*, *mid*+1, *right*) (järjestetään oikea puoli)
- 5 Merge(*A*, *left*, *mid*, *right*) (lomitetaan järjestetyt puolikkaat yhteen)

Insertion-Sort(A)

```
1 for next_elem := 2 to A.length do
2   key := A[next_elem]
3   place := next_elem - 1
4   while place > 0 and A[place] > key do
5     A[place + 1] := A[place]
6     place := place - 1
7   A[place + 1] := key
```

(siirretään osien välistä rajaa)
(otetaan uusi alkio käsittelyyn)

(etsitään uudelle alkiolle oikea paikka)
(raivataan uudelle alkiolle tilaa)

(asetetaan uusi alkio oikealle paikalleen)

STL:n algoritmien ja säiliöiden mukauttaminen

COMP.CS.300 Tietorakenteet ja algoritmit 1

Matti Rintala (matti.rintala@tuni.fi)

STL:n mukauttaminen

- Monet algoritmit ottavat parametrinaan lambdan/funktion, jota algoritmi käyttää (=kutsuu)
- Esim. sort ja alkioiden vertailu, find_if ja millaista etsitään...
- Joidenkin säiliöiden toimita myös mukauttavissa
- Esim. map ja avainten järjestys

Quicksort(A , $left$, $right$)

- 1 **if** $left < right$ **then** (triviaalitapaukselle ei tehdä mitään)
- 2 $pivot := \text{Partition}(A, left, right)$ (jaetaan pieniin ja suuriin, $pivot$ jakokohta)
- 3 Quicksort(A , $left$, $pivot-1$) (järjestetään jakoalkiota pienemmät)
- 4 Quicksort(A , $pivot+1$, $right$) (järjestetään jakoalkiota suuremmat)