

# Satunnaistaminen (Randomization)

***COMP.CS.300 Tietorakenteet ja algoritmit 1***

***Matti Rintala (matti.rintala@tuni.fi)***

# Ongelma

- Pahin vs. "keskimääräinen" tapaus
- Tapausten todennäköisyydet
- Todellinen data usein painottunut

# Satunnaistaminen

- Ratkaisu: "sotketaan" datan järjestys alussa tai algoritmin aikana
- Tulos: kaikki tapaukset yhtä todennäköisiä
- Pahin tapaus ei enää yleisempi kuin muut
- Satunnaistaminen vie myös aikaa!
- **Pahin tapaus edelleen yhtä paha!**

# Satunnaistettu pikalajittelu

## Randomized-Quicksort( $A$ , $left$ , $right$ )

- 1 **if**  $left < right$  **then**
- 2      $pivot := \text{Randomized-Partition}(A, left, right)$
- 3     Randomized-Quicksort( $A$ ,  $left$ ,  $pivot-1$  )
- 4     Randomized-Quicksort( $A$ ,  $pivot+1$ ,  $right$ )

## Randomized-Partition( $A$ , $left$ , $right$ )

- 1  $pivot := \text{Random}(left, right)$  (valitaan satunnainen vertailualkio)
- 2  $A[ right ] \rightleftharpoons A[ pivot ]$  (vaihdetaan pivot viimeiseksi)
- 3 **return** Partition( $A$ ,  $left$ ,  $right$ ) (normaali ositus)

# Satunnaistettu haku?

## Randomized-Search(*A*, *value*)

```
1 Shuffle(A)                (sekoita A:n järjestys)
2 for i in 1..A.size        (käy läpi taulukon indeksit)
3     if A[i] = value then
4         return i
```