

Föreläsning 16: Sortering

- Insättningsortering
- Samsortering

Insättningsortering (Insertion sort)

Bäst av dom enkla metoderna är insättning. Den känns särskilt naturlig om man hämtar talen ett efter ett från en fil och sorterar in dom i en lista.

- Jämför nya talet med sista talet i listan.
- Om nya är större, lägg det sist i listan.
- Annars, putta ner (bak) sista talet ett pinnhål och jämför igen.
- Putta ner (bak) så många tal som behövs för att sätta in nya talet på rätt plats.
- Upprepa för varje nytt tal.

Insättning fungerar också om talen finns i listan från början. Efter k insättningar är vektorsegmentet $v[1] \dots v[k]$ ordnat och $v[k+1]$ är det nya talet som ska sättas in.

Komplexiteten är i allmänhet $O(N^2)$ men om vektorn är nästan sorterad från början är insättning den snabbaste algoritmen.

Samsortering (Merge sort)

Om man har flera sorterade småfiler är det lätt att *samsortera* dom till en fil. Det här kan man också göra med en lista om man har extrautrymme för att kopiera den till två andra hälften så långa listor. Det här ger en rekursiv tanke!

- Dela listan i två hälften så långa listor.
- Sortera varje halva för sej.
- Samsortera till ursprungliga listan.

Komplexiteten blir $O(N \log N)$, lika snabb som quicksort men kräver extra minnesutrymme.