

# Sorteringsalgoritmer

## Information till läraren

### Mål med problemet

Att eleverna ska få lära sig övergripande teori om olika sorteringsalgoritmer och hur man kan implementera någon eller några av de mer vanliga sorterna.

### Förkunskaper

Man bör känna sig bekväm med hantering av fält (arrays) och loopar.

# Övergripande upplägg

## Introduktion

Presentera problemet och eventuellt en förutbestämd lista som ska sorteras.

Här är några exempel:

[1, 2, 3, 4, 5, 7, 6]

[9, 8, 3, 6, 4, 2, 7, 1, 5, 0]

## Genomförande

Eleverna ska, förslagsvis i grupper om två (alternativt 3 i en grupp), implementera en eller flera sorteringsalgoritmer (behöver inte begränsas till de som vi har använt oss av i lösningsförslaget) som sorterar en given lista.

## Diskussion

Diskutera i helklass:

- Lyckades ni sortera listan?
- Vilka sorteringsalgoritmer använde ni?
  - Tror ni, givet en mycket längre lista, att någon är snabbare än de andra?

# Ytterligare information

## Bakgrund och lösningsförslag

Att sortera data är väldigt centralt i de flesta system, och varje programmerare förväntas ha koll på hur många av de vanligare sorteringsalgoritmerna fungerar och när man bör välja den ena framför den andra.

Lösningsförslag för bubble sort och insertion sort: <https://pastebin.com/8AM8Nnue>

## Användningsområden:

För att sortera varor i webbshoppar efter kategori, pris etc. Kontakter i din mobil är t.ex ofta sorterade i bokstavsordning, och länkarna som dyker upp när du googlar något efter hur många andra sidor som refererar till dem.

## Förslag på vidareutveckling

Sorteringsalgoritmer har så kallade komplexiteter, ett sätt att beskriva hur snabbt det tar att utföra en algoritm i olika fall. Som man kanske kan gissa så finns det sorteringsalgoritmer som i många fall är bättre än andra. En av dessa är merge sort, som använder sig av så kallad "divide and conquer"-metoden. Det innebär att listan delas upp i mindre delar som sorteras var och en för sig, och sedan sammanförs. Detta är alltså en av de många metoder som man kan använda för att effektivisera sorteringen.