

# Sortera en kortlek

En introduktion till sorteringsalgoritmer

## Information till läraren

### Mål med uppgiften

Uppgiften går ut på att lära sig sorteringsalgoritmer specifikt, men målet är också att på ett övergripande plan förstå hur man kan ge instruktioner till en dator samt förstå begreppet algoritm och hur en sådan kan konstrueras.

### Förkunskaper

Inga speciella förkunskaper krävs.

### Material

I uppgiftsförslaget används ett antal kortlekar. Vi rekommenderar att man har en kortlek per 8 elever. De bör också sorteras i färg innan lektionen.

### Bakgrund

Det är bra om du som lärare har en viss förkunskap om sorteringsalgoritmer. Vi skickar även med en kort text om detta.

# Övergripande upplägg

## Introduktion

Förklara vad en algoritm är.

- Vad tror ni att man kan använda sorteringsalgoritmer till?
- Vad för typer av kommandon tror ni att man kan ge till en dator när det gäller sortering?
- Vad krävs för att en lista ska kunna vara sorterbar?

## Genomförande

I grupper om 2 (alternativt 3 i en grupp), förslagsvis med en tidsbegränsning på ca 15-20 minuter. Varje grupp får en färg utav en kortlek. Eleverna får nu i uppgift att sortera sina kort enligt följande regler (Vi skickar även med en powerpoint med reglerna):

1. 10 kort placeras med baksidan uppåt.
2. Du får vända på två kort i taget, jämföra dem och om du vill byta plats på dem.
3. Du kan inte minnas vad för värde ett kort som du inte tittar på (dvs ett nedåtvänt kort) har, men du får hålla koll på om kortet är sorterat eller inte.

När du anser dig ha sorterat alla kort så får du vända på alla kort och kolla om du har gjort rätt.

När de kommit fram till en algoritm för att sortera korten så får de i uppgift att skriva ner de steg de gör. Därefter ska de byta sina instruktioner med en annan grupp och de ska kunna sortera sin kortlek genom att enbart följa instruktionerna.

## Diskussion

Eleverna har nu alltså försökt att komma på egna sorteringsalgoritmer

- Fungerar de?
  - Om inte, vad gick fel?
- Hur gick det att följa instruktionerna? (Var det svårt/lätt?)
- Hur gick det att skriva instruktioner? (Var det svårt/lätt?)
- Hur kändes det att försöka tänka som datorer?

# Ytterligare information

## Bakgrund

Vi har konstruerat uppgiften så att du som lärare inte ska behöva några förkunskaper om sortering eller programmering. Istället har vi skrivit ihop en kort sammanfattning med allt du behöver veta innan du genomför uppgiften.

En dator kan inte som människor se vilket kort som är lägst genom att titta på alla korten samtidigt. Istället behöver den jämföra korten två och två, vilket vi här illustrerar med att man har korten med baksidan upp, och bara får titta på två kort i taget. Ett sätt att sortera korten är nu att titta på de två första korten, och byta plats på dem om de ligger i fel ordning. Därefter vänder man det andra och tredje kortet och gör samma sak, osv. På så sätt förflyttar du det största kortet längst ut till höger. Därefter upprepar du samma sak, men nu utan det sista kortet. På så sätt lägger du för varje upprepning, eller iteration, ett kort på rätt plats. Detta kallas "bubble sort".

Tanken är att eleverna själva ska få komma på sorteringsalgoritmer, så din uppgift är att presentera uppgiften och finnas som stöd. Eleverna får alltså vända på två kort i taget, titta på dem och eventuellt byta plats på dem (dessa regler ska efterlikna hur en dator fungerar. De skulle kunna sortera korten enligt det som nämns ovan, eller på ett annat sätt. Det viktiga är att det fungerar. Tanken är alltså INTE att man ska komma ihåg vart korten ligger, utan man ska se det som att det bara är de som man just då tittar på som man känner till. När man kommit på sin algoritm ska man försöka skriva ner de steg man gör, på ett sådant sätt att en annan grupp kan använda dessa instruktioner för att sortera sin kortlek. Instruktionerna ska alltså inte vara specifika för en viss ursprunglig blandning, utan fungera oberoende av startläge.

Ett tips för elever som tycker det är svårt kan vara att de kan utgå från hur de i vanliga fall skulle sortera en kortlek. När de kommit på det kan de försöka bryta ner detta i mindre steg som följer dessa regler.

## Förslag på hur uppgiften kan presenteras

En algoritm är en process för att lösa ett specifikt problem bestående av flera steg som utförs efter varandra. Ett enkelt exempel kan vara att du rostar bröd genom att först stoppar ner brödet i brödrosten, sedan trycker du på start knappen och sist väntar du på att maskinen ska bli färdig. Sen tar du ut brödet och äter det.

En dator är väldigt bra på att utföra små enkla steg väldigt snabbt och därför är de väldigt bra på att lösa problem med hjälp av algoritmer. Problemet är att vi måste ge väldigt specifika instruktioner till datorn vad den ska göra, och sådana saker som vi anser vara väldigt enkla kan vara komplicerat att förklara för en dator.

I dagens lektion ska vi fokusera på en specifik typ av algoritmer, de så kallade sorteringsalgoritmerna. Som ni kanske hör på namnet används de för att sortera saker.

- Vad tror ni att man kan använda sorteringsalgoritmer till?
  - Sortera en lista av namn
  - Organisera en spellista för musik
  - Ranka resultat av en google sökning
  - Visa nyheter i kronologisk ordning
  - Hitta medianer
  - Hitta närmaste par
  - Sökning i en databas
  - Hitta dubletter i en maillista
  - Komprimera data
  - Datorgrafik
- Vad för typer av kommandon tror ni att man kan ge till en dator när det gäller sortering?
  - Jämföra två kort.
  - Gå ett steg till höger/vänster.
  - Byta plats på två kort.
  - Minnas att den tittar på ett kort.
  - Säga till att den är färdig.
  - (Hålla koll på flera kortlek/lor och flytta kort mellan dem.)
  - (Veta vilken position ett visst kort har i listan.)
- Vad krävs för att en lista ska vara sorteringsbar?
  - Elementen ska gå att jämföras med varandra

De sista två är för något mer avancerade algoritmer.

## Förslag på fördjupning

- En idé på vidareutveckling är att de får leta reda på en eller flera existerande sorteringsalgoritmer, genom att till exempel använda wikipedia, och sedan skriva instruktioner för hur dessa skulle användas för att sortera en kortlek.
- Skriv ner hur många jämförelser som ni gör i era algoritmer. Försökt lista ut hur många som kommer göras i värsta fallet.