

Sortering

Programmering 1

vt 23

Innehåll

Sorteringsalgoritmer

Aktivitet

Komplexitet

Sorteringsalgoritmer

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Övningar

Vad är en sorteringsalgorithm?

- ▶ En sorteringsalgorithm är ett system för att sortera element
- ▶ Varierande komplexitet
- ▶ Varierande hastighet
- ▶ Varierande minnesanvändning

Varför ska vi lära oss det här?

- Det finns färdiga funktioner. Så varför?

Varför ska vi lära oss det här?

- ▶ Det finns färdiga funktioner. Så varför?
- ▶ Förståelse: det krävs mycket kunskap för att genomföra
- ▶ Utmanande: många algoritmer är tekniskt svåra

Innehåll

Sorteringsalgoritmer

Aktivitet

Komplexitet

Sorteringsalgoritmer

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Övningar

Aktivitet

Sortera kort

- ▶ Du kommer att få tretton kort, Ess–Kung (1–13)
- ▶ Blanda korten
- ▶ Sortera korten i storleksordning
- ▶ Hur gjorde du?
- ▶ Varför gjorde du så?

Min gissning på er algoritm

Insertion sort

1. Första kortet är "rätt"
2. lägg på rätt sida om första
3. stoppa in på rätt plats, bland de ordnade
4. stoppa in på rätt plats, bland de ordnade
5. ...

Innehåll

Sorteringsalgoritmer

Aktivitet

Komplexitet

Sorteringsalgoritmer

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Övningar

Komplexitet

- ▶ Tidskomplexitet handlar förenklat om hur många jämförelser som behöver göras
- ▶ Om du loopar igenom en lista n -gänger är tidskomplexiteten $O(n)$
- ▶ Har du en loop i en loop och loopar igenom n gånger för varje element i listan är tidskomplexiteten $O(n^2)$
- ▶ Minneskomplexitet är hur mycket datorminne som går åt.
- ▶ Man brukar mäta två två sätt:
 1. Total minnesåtgång
 2. Extra minnesåtgång
- ▶ Här nedan kommer exemplena bara vara med extra åtgång.
- ▶ Skapar man ingen extra lista för att sortera så är komplexiteten $O(1)$

Innehåll

Sorteringsalgoritmer

Aktivitet

Komplexitet

Sorteringsalgoritmer

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Övningar

Olika sorteringsalgoritmer

- ▶ Bubble Sort
- ▶ Insertion Sort
- ▶ Selection Sort
- ▶ Quick Sort

Bubble Sort

- ▶ Intuitiv algoritm
 - ▶ Lätt att implementera
 - ▶ Tidskomplexitet: $O(n^2)$ (våldigt långsam)
 - ▶ Minneskomplexitet: $O(1)$ (minneseffektiv)
1. Jämför första elementet med sin granne till höger
 2. Ordna de två elementen i storleksordning
 3. Jämför nästa två element o.s.v.
 4. Upprepa för alla element i listan

Insertion Sort

- ▶ Intuitiv algoritm
 - ▶ Tidskomplexitet: $O(n^2)$ (våldigt långsam)
 - ▶ Minneskomplexitet: $O(1)$ (minneseffektiv)
1. Anta att första elementet är rätt
 2. Bubble sorta de två första elementen
 3. Bubble sorta tills det tredje är på rätt plats
 4. Upprepa för alla element i listan

Selection Sort

- ▶ Intuitiv algoritm
 - ▶ Tidskomplexitet: $O(n^2)$ (våldigt långsam)
 - ▶ Minneskomplexitet: $O(1)$ (minneseffektiv)
1. Hitta det största elementet i listan
 2. Placera det sist
 3. Hitta näst största elementet och placera näst sist
 4. Upprepa för alla element i listan

Quick Sort

- ▶ Rekursiv algoritm
 - ▶ Tidskomplexitet: $nO(n)$ (våldigt långsam)
 - ▶ Minneskomplexitet: $O(1)$ (minneseffektiv)
1. Välj ett element i listan
 2. Placera alla element mindre än det valda före och alla större efter
 3. Välj ett element ur den nedre halvan och upprepa
 4. Välj ett element ur övre halvan och upprepa

Innehåll

Sorteringsalgoritmer

Aktivitet

Komplexitet

Sorteringsalgoritmer

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Övningar

Övningar

1. Implementera Bubble Sort
2. Implementera Insertion Sort
3. Implementera Selection Sort
4. Implementera Quick Sort
5. Implementera Stalin Sort