

Föreläsning 2 i ADK

Repetition av sortering

Stefan Nilsson

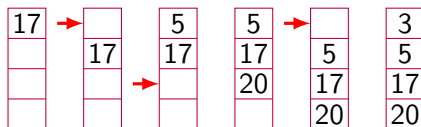
KTH

Insättningssortering

Algoritm:

- Placera in första elementet på första platsen
- För varje nytt element x som ska sorteras in
 - Leta reda på var x ska in
 - Förskjut alla element till höger om den platsen ett steg
 - sätt in x

Exempel, sortera in talen 17 5 20 3:



Antal operationer: $\mathcal{O}(n^2)$

Minnesåtgång: 0 extra element

	Medeltal	Värsta fallet
Jämförelser	$\frac{n^2}{4}$	$\frac{n^2}{2}$
Flyttar	$\frac{n^2}{4}$	$\frac{n^2}{2}$

Urvalssortering

Algoritm:

- Välj ut det minsta elementet
- Byt plats på minsta och första elementet
- Fortsätt på samma sätt med resten av elementen

Exempel:

4	3	3	3	3
7	7	4	4	4
4	4	7	4	4
19	19	19	19	7
3	4	4	7	19

Antal operationer:

- $\frac{n(n-1)}{2}$ dvs $\mathcal{O}(n^2)$ jämförelser
- $n - 1$ platsbyten
- Detta gäller oberoende av hur indata är ordnat

Minnesåtgång: 1 extra element

Mergesort

Algorithm:

```
function MERGESORT( $v[i..j]$ )  
  if  $i < j$  then  
     $m \leftarrow \lfloor \frac{i+j}{2} \rfloor$   
    MERGESORT( $v[i..m]$ )  
    MERGESORT( $v[m+1..j]$ )  
    MERGE( $v[i..j]$ ,  $v[i..m]$ ,  $v[m+1..j]$ )
```

Analys:

- Låt $T(n)$ = Tiden att sortera n tal med mergesort
- $$T(n) = \begin{cases} \Theta(1) & \text{om } n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + \Theta(n) & \text{om } n > 1 \end{cases}$$
- Om $n = 2^m$ får vi $T(n) = \begin{cases} \Theta(1) & \text{om } n = 1 \\ 2T(\frac{n}{2}) + \Theta(n) & \text{om } n > 1 \end{cases}$
- "Master theorem": Eftersom $n^{\log_2 2} = n^1 \in \Theta(n)$
så är $T(n) = \underline{\Theta(n \log n)}$

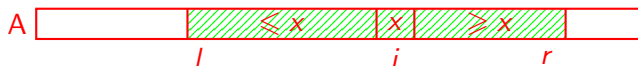
Minnesåtgång: n extra element (för merge)

Quicksort - $\mathcal{O}(n \log n)$ i genomsnitt

Algorithm som sorterar arrayen $A[l..r]$:

- 1 Partitionera A efter ett godtyckligt element x , dvs ordna om A så att:
 - Alla element som är $< x$ kommer till vänster
 - Alla element som är $> x$ kommer till höger

Låt i vara indexet för platsen i A där x hamnar



- 2 Anropa Quicksort rekursivt på den vänstra delen av A (från index l till $i - 1$)
- 3 Anropa Quicksort rekursivt på den högra delen av A (från index $i + 1$ till r)

När ska rekursionen avbrytas?

- **Alternativ 1:** Då $l \geq r$, d.v.s. då högst ett element finns i arrayen.
- **Alternativ 2:** Då $r - l < 10$ d.v.s. då högst tio element finns i arrayen.
Sortera de återstående elementen med insättningssortering.