Søgning og Sortering

- Søgning
 - Linæer søgning
 - Binær søgning
- Sortering
 - Indsættelsessortering
 - Flettesortering

Søgning og Sortering

- Søgning
 - Linæer søgning
 - Binær søgning
- Sortering
 - Indsættelsessortering
 - Flettesortering

Søgning

- Søgning. Givet en sorteret tabel A og et tal x, afgør om der findes indgang i, så A[i] = x.
- Sorteret tabel. En tabel A[0..n-1] er sorteret hvis A[0] \leq A[1] \leq \cdots \leq A[n-1] (ikkefaldende rækkefølge).

_				_											14
	1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Linæer søgning

- Lineær søgning. Undersøg for alle indgange i om A[i] = x.
- Tid. Θ(n)
- Udfordring. Kan vi udnytte at tabellen er sorteret til at gøre det bedre?

_				_											14
	1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

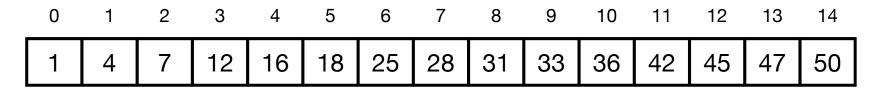
Binær søgning

- Binær søgning (binary search). Kig på midterste indgang m i A.
 - hvis A[m] = x returner sand og stop.
 - hvis A[m] < x fortsæt rekursivt på højre halvdel.
 - hvis A[m] > x fortsæt rekursivt på venstre halvdel.
- Stop hvis tabellen har størrelse ≤ 0 og returner falsk.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Binær søgning

```
BINÆRSØGNING(A,i,j,x)
  if j < i return false
  m = Li+j/2
  if A[m] = x return true
  elseif A[m] < x return BINÆRSØGNING(A,m+1,j,x)
   else return BINÆRSØGNING(A,i,m-1,x) // A[m] > x
```



- Tid. Hvor hurtigt kører den?
- Analyse 1. Analog til analyse af rekursiv toppunktsalgoritme.
 - Et rekursivt kald tager konstant tid.
 - Hvert rekursivt kald halverer tabellen vi kigger på. Vi stopper når tabellen har størrelse ≤ 0.
 - \Rightarrow Køretiden er $\Theta(\log n)$

Binær søgning

- Analyse 2. Lad T(n) være køretiden for binær søgning.
 - Opskriv og udregn rekursionsligningen for T(n).

$$T(n) = \begin{cases} T(n/2) + c & \text{hvis } n > 1\\ d & \text{hvis } n = 1 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + c$$

$$= T\left(\frac{n}{4}\right) + c + c$$

$$= T\left(\frac{n}{8}\right) + c + c + c$$

$$\vdots$$

$$= T\left(\frac{n}{2^k}\right) + ck$$

$$\vdots$$

$$= T\left(\frac{n}{2^{\log_2 n}}\right) + c\log_2 n$$

$$= T(1) + c\log_2 n$$

$$= d + c\log_2 n$$

$$= \Theta(\log n)$$

Søgning

- Vi kan søge i en sorteret tabel i
 - Θ(n) tid med lineær søgning.
 - Θ(log n) tid med binær søgning.