Föreläsning 14 i ADK

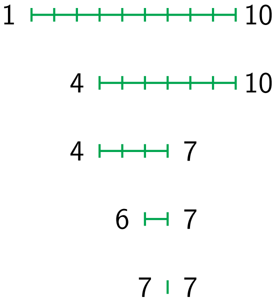
Undre gränser

Viggo Kann

KTH

Talgissning med tuff motståndare

```
Gissa ett tal mellan 1 och 10
   3
Större!
   8
Mindre!
   5
Större!
   6
Större!
Du klarade det inte på 4 gissningar
Rätt svar var 7
```



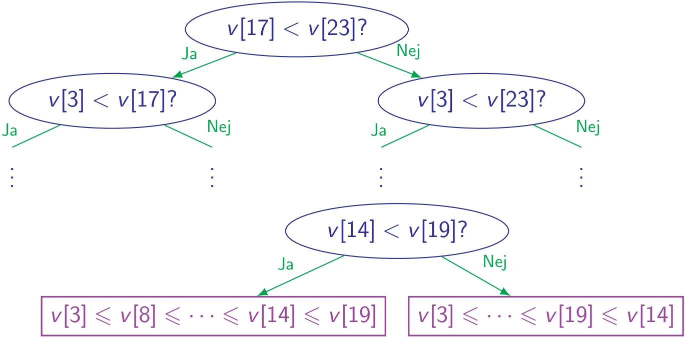
Talgissning med tuff motståndare

```
gissning ← 0
min ← 1
max ← 10
WRITE("Gissa ett tal mellan " min " och " max)
while min < max do
    gissning ← gissning+1
    x ← READINTEGER( )
    m ← (min+max)/2
    if x < m then
        WRITE("Större!")
    else
        WRITE("Mindre!")
    if x < m and x ⩾ min then min ← x+1
    if x ⩾ m and x ⩽ max then max ← x-1
WRITE("Du klarade det inte på " gissning " gissningar")
WRITE("Rätt svar var " min)
```

Hur många jämförelser krävs minst för att sortera n element?

$$A : \{n \text{ element}\} \to \{n\text{-permutationer}\}$$

- Anta att den enda operationen (förutom tilldelning) som kan göras på element är jämförelse mellan två element
- Då kan algoritmen A beskrivas som ett **beslutsträd:**



Hur många jämförelser krävs minst för att sortera n element?

- Trädets höjd $=$ tidskomplexiteten
- Antalet löv $\geqslant$ antalet permutationer $= n!$
- Ett binärträd av höjd h har högst 2^h löv
- $\Rightarrow$ tidskomplexiteten är $\Omega(n \log n)$

Komplexitet för medianproblemet

Problem: Hitta medianen bland n element

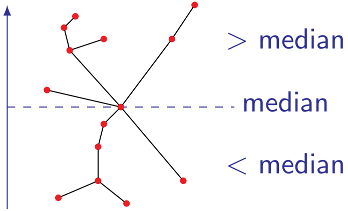
Hur många jämförelser krävs för att hitta medianelementet i värsta fallet?

Trivial övre gräns:

- Följande algoritm gör $\mathcal{O}(n \log n)$ jämförelser:
- $\textsc{Mergesort}(S[1..n])$; **return** $S[\frac{n+1}{2}]$

Trivial undre gräns:

- Varje algoritm som hittar medianen måste göra en kedja av jämförelser mellan varje annat tal och medianen:



- Alltså krävs minst $n - 1$ jämförelser

Undre gräns för tuff motståndare

- Idé: Visa att en tuff motståndare kan tvinga algoritmen att göra onyttiga jämförelser
- Jämförelsen $x > y$ är onyttig om $x >$ medianen och $y <$ medianen

Taktik för tuffa motståndaren:

- Välj 0 som median
- Tilldela element ett värde först då algoritmen vill jämföra det med något annat element

Jämförelse mellan	Svara	Tilldela
Nytt och nytt	$>$	Första positivt, andra negativt
Positivt och nytt	$>$	Negativt tal
Negativt och nytt	$<$	Positivt tal
Känt och känt	Sanningen	-

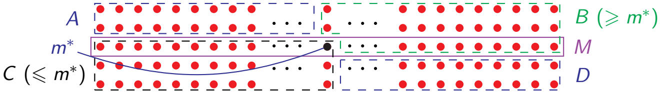
Undre gräns för tuff motståndare

- När $\frac{n-1}{2}$ negativa eller $\frac{n-1}{2}$ positiva värden har tilldelats har inte motståndaren något val längre.
- Varje algoritm kan på så sätt tvingas göra $\frac{n-1}{2}$ onyttiga jämförelser utöver dom $n-1$ nyttiga: Vi får undre gränsen $n-1+\frac{n-1}{2}=\frac{3n-3}{2}$ jämförelser.
- Bästa kända undre gränsen är $2{,}01n$ [Dor, Håstad, Ulfberg, Zwick 2001]

Hitta i-te minsta talet i S

function $\textsc{Select}(S[1..n], i)$

- **if** $n \leqslant 5$ **then**
  - $\textsc{Sort}(S[1..n])$
  - **return** $S[i]$
- Dela upp S i $\frac{n}{5}$ grupper med 5 tal i varje.
- Hitta medianen i varje grupp och partitionera gruppen efter den
- $M \leftarrow \{medianer\}$
- $m^* \leftarrow \textsc{Select}(M, \frac{n/5}{2})$



- $S_1 \leftarrow C \cup \{x \in A \cup D : x \leqslant m^*\}$
- $S_2 \leftarrow B \cup \{x \in A \cup D : x > m^*\}$
- **if** $i = |S_1|$ **then return** m^*
- **if** $i < |S_1|$ **then return** $\textsc{Select}(S_2, i - (n - |S_2|))$
- **else return** $\textsc{Select}(S_2, i - (n - |S_2|))$

Hitta i-te minsta talet i S

Komplexitetsanalys: (Antal jämförelser)

- $T(n) \leqslant 6 \cdot \frac{n}{5} + T(\frac{n}{5}) + 2 \cdot \frac{n}{5} + T(\frac{n}{5} + \frac{n}{2}) = 1.6n + T(0.2n) + T(0.7n)$
  - $6 \cdot \frac{n}{5}$: Hitta medianerna i grupperna
  - $T(\frac{n}{5})$: Hitta m^*
  - $2 \cdot \frac{n}{5}$: Fördela på S_1 och S_2
  - $T(\frac{n}{5} + \frac{n}{2})$: Rekursion på S_1 eller S_2
- Man kan visa att $T(n) \leqslant 16n$
- Detta ger övre gränsen $16n$ för medianproblemet
- Bästa kända övre gränsen är $2.95n$ [Dor, Zwick 1999]