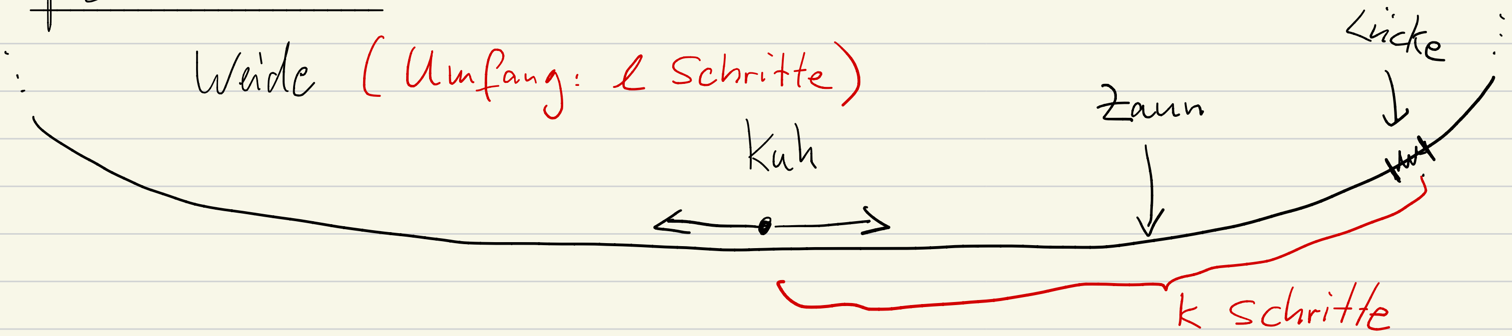
AD20-02

"pasture break"



Kurzichtige Kuh sucht Lücke im Zaun

Ziel: möglichst wenig Schritte

Algorithmus 1: gehe in eine Richtung bis Lücke gefunden

Fall "richtige Richtung": k Schritte

Fall "falsche Richtung": $\ell - k$ Schritte

schlimmstenfalls: $\ell/2 \leq \max\{k, \ell - k\} \leq \ell$ Schritte

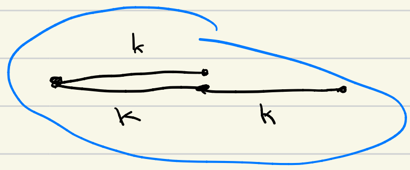
"worst-case analysis"

geht es besser? (wenn ℓ viel grösser als k)

Algorithmus 2: k Schritte links, $2k$ Schritte rechts

Fall "Lücke links": k Schritte

Fall "Lücke rechts": $3k$ Schritte



worst-case: $3k \leq O(k)$ Schritte

wenn k unbekannt?

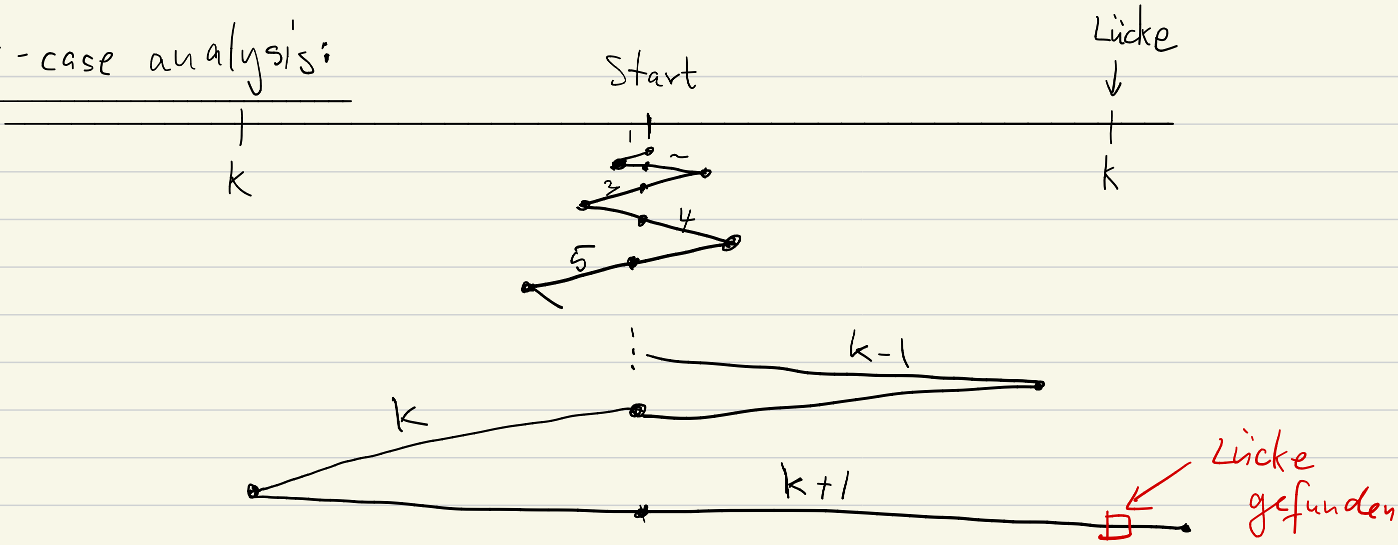
Algorithmus 3: 1 links, zurück zum Start,

2 rechts, —— " ——,

3 links, —— " ——

⋮

worst-case analysis:



Schritte:

Induktion $\underbrace{2\cdot 1 + 2\cdot 2 + \dots + 2\cdot k} + k$

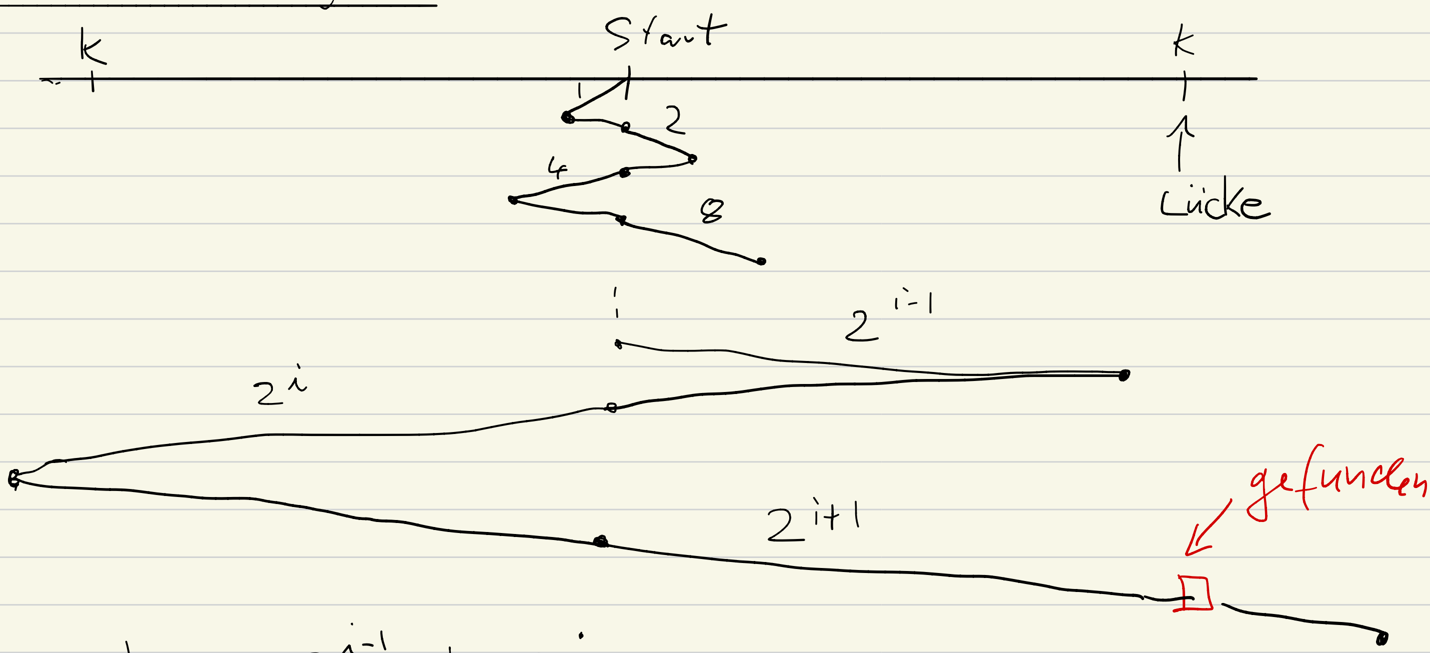
$= k\cdot(k+1) \qquad + k$

$= k\cdot(k+2) \le O(k^2)$

geht es besser?

Algorithmus 4: verdoppeln anstatt um 1 erhöhen

worst-case analysis:



wobei $2^{i-1} < k \le 2^i$

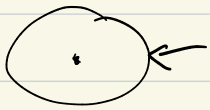
Schritte: $2\cdot(\underbrace{1 + 2 + 4 + \dots + 2^{i-1} + 2^i}_{\text{Induktion}}) + k$

$= 2\cdot(\underbrace{2^{i+1} - 1}_{< 4k}) + k < 9k$

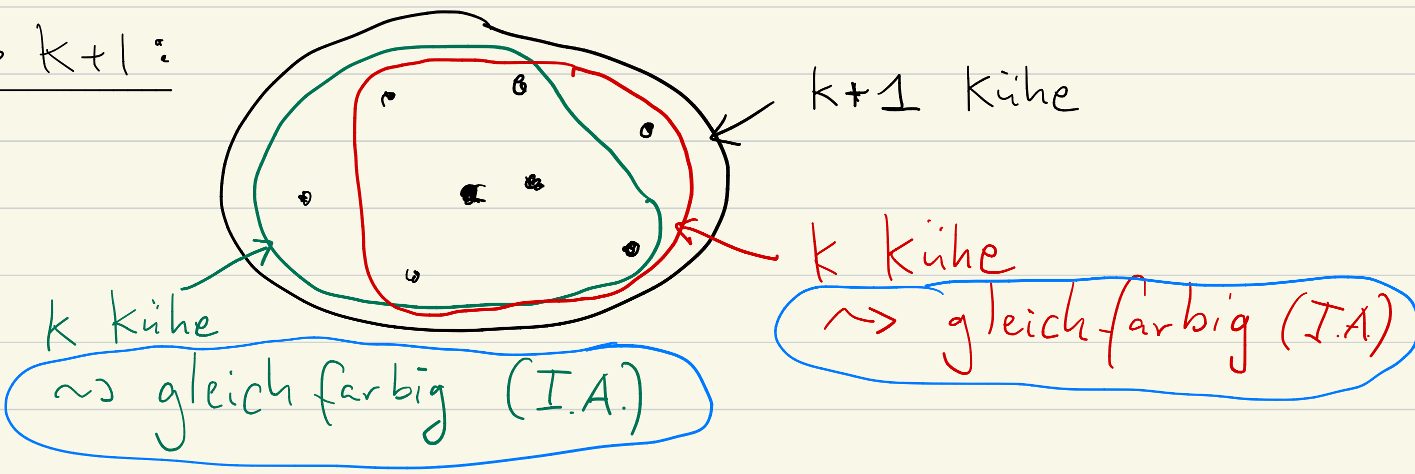
Einschub: Tücken bei Induktionsbeweise

Aussage $A(k)$: in jeder Herde von k Kühen haben alle diesselbe Farbe

beweise per Induktion: $A(k)$ gilt für alle $k \in \mathbb{N}$, $k \geq 1$?

$k=1$:  1 Kuh ✓

$k \to k+1$:



beide k-Herden teilen sich Kühe

~> alle $k+1$ Kühe gleiche Farbe

Fehler: "$k \to k+1$" Beweis falsch für $k=1$

Star Suche

Problem: finde "Star" unter n Personen

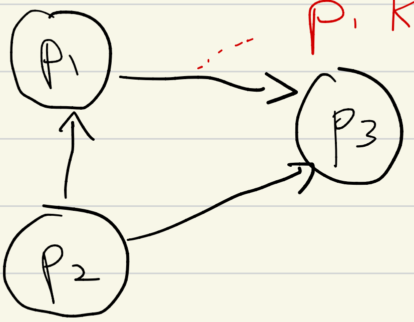
$$p_1, \dots, p_n \qquad (n \geq 2)$$

Definition: p_s ist Star falls

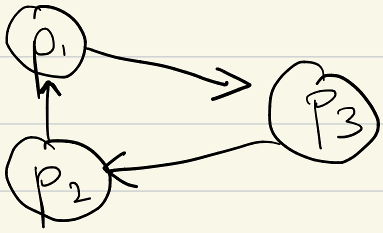
- jede andere Person kennt p_s
- p_s kennt keine andere Person

Beispiele:

"p_1 kennt p_3"



Star: p_3



kein Star

immer ≤ 1 Star

elementare Operation: frage $p_i \xrightarrow{?} p_j$

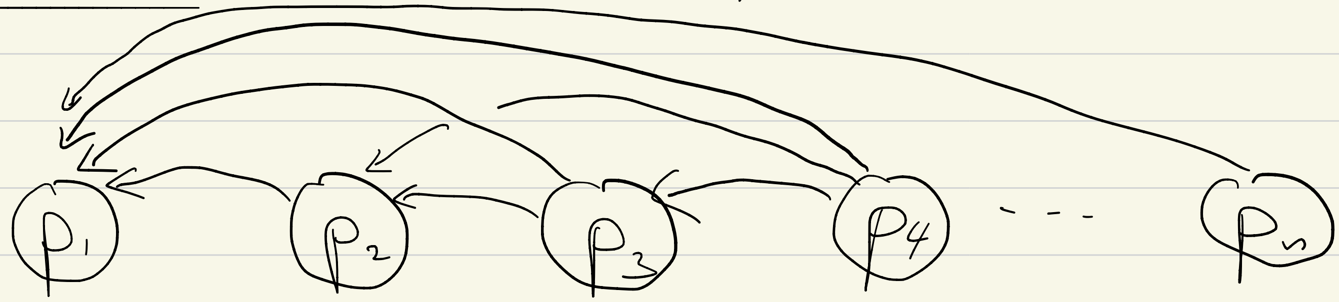
naiver Algorithmus: alle $n \cdot (n-1)$ Fragen

geht es besser?

rekursiver Ansatz: $(n > 2)$

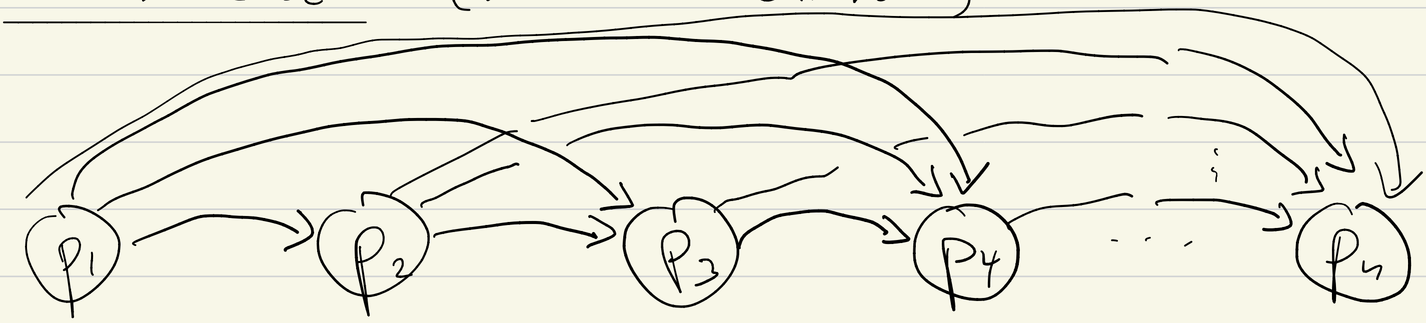
1. Suche Star p_s unter $p_1, \dots, p_{n-1}$ rekursiv
2. Teste ob p_s Star für $p_1 \cdots p_n$: 2 Fragen
3. Teste ob p_n ——" —— : $2(n-1)$ Fragen

best case: (nie Schritt 3)



n	2	3	4		n	Total
Fragen	2	+2	+2		+2	$2 \cdot (n-1)$ $\leq O(n)$

worst case: (immer Schritt 3)



n	2	3	4	...	n	Total
Fragen	2	$+2 \cdot 2$	$+2 \cdot 3$	...	$+2 \cdot (n-1)$	$n \cdot (n-1)$ $\leq O(n^2)$

geht es besser?

Idee: stelle sicher dass p_n kein Star

$\rightsquigarrow$ Schritt 3 entfällt

vor Schritt 1:

vertausche p_{n-1} und p_n falls $p_{n-1} \to p_n$

(1 Frage)

<u>Rekurrenz:</u>

$$F(n) \leq 3 + F(n-1)$$

$$F(2) = 2$$

wiederholtes Einsetzen:

$$F(n) = 3 + F(n-1)$$

$$= 3 + 3 + F(n-2)$$

$$\vdots$$

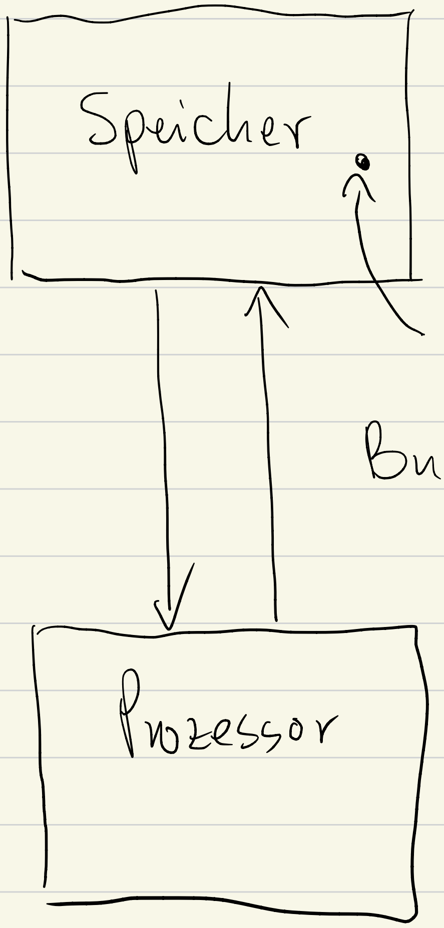
$$= \underbrace{3 + \cdots + 3}_{n-2} + F(2)$$

$$= 3(n-2) - \quad = \quad n - 4 \leq O(n)$$

nicht besprochen:

Standard Rechner modell:

gute Wahl von elementaren Operation
(für fast alle Berechnungs probleme)



Liste von Speicherzellen
- beliebig viele
- frei addressier bar

ite Speicherzelle

Bus (verbindet Speicher und Prozessor)

Führt elementare Operationen aus

- Lesen / Schreiben von Speicher zellen
- Vergleichen $(=, >, <)$
- Arithmetik $(+, -, \cdot, /)$

- Laufzeit
- worst-case Laufzeit