

Übung 6 – Algorithmen II

Yaroslav Akhremtsev, Demian Hespe – yaroslav.akhremtsev@kit.edu, hespe@kit.edu

Mit Folien von Michael Axtmann (teilweise)

http://algo2.iti.kit.edu/AlgorithmenII_WS17.php

Institut für Theoretische Informatik - Algorithmik II

```
    result = current_weight;
    return true;
}

for( EdgeID eid = graph.edgeBegin( current ); eid != graph.edgeEnd( current ); ++eid ){
    const Edge & edge = graph.getEdge( eid );
    COUNTING( statistic_data.inc( DijkstraStatisticData::TOUCHED_EDGES ); )
    if( edge.forward ){
        COUNTING( statistic_data.inc( DijkstraStatisticData::RELAXED_EDGES ); )
        weight new_weight = edge.weight + current_weight;
        GUARANTEE( new_weight >= current_weight, std::runtime_error, "Weight overflow detected." );
        if( !priority_queue.isReached( edge.target ) ){
            COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_EDGES ); )
            COUNTING( statistic_data.inc( DijkstraStatisticData::REACHED_NODES ); )
            priority_queue.push( edge.target, new_weight );
        } else {
            if( priority_queue.getCurrentKey( edge.target ) > new_weight ){
                COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_NODES ); )
                priority_queue.decreaseKey( edge.target, new_weight );
            }
        }
    }
}
```

■ Online Algorithmen

(Probleme mit beschränkter Information gut abschätzen)

- Information kommt **nach und nach** an
- Entscheidungen / Berechnungen mit **beschränkter Information**
 - normalerweise **suboptimale Lösungen**

Beispiele

- *Ski Rental Problem*
- Job-Scheduling / Memory-Paging
- *Streaming Algorithmen*
- Suche in Labyrinthen / Routing in Kommunikationsnetzen
- ...

- Ein Online Algorithmus ALG hat einen kompetitiven Faktor

$$c = \sup_I \frac{ALG(I)}{OPT(I)}$$

mit OPT optimaler Offline Algorithmus, I Probleminstanz
(hier für Minimierungsprobleme; "Approximationsfaktor" c)

Einige Eigenschaften

- $c = \infty$ möglich $\rightarrow$ Online Algorithmus beliebig schlecht
- $c = \text{const.}$ (normal nicht von Problemgröße abhängig)
aber ev. abhängig von weiteren Problemparametern

Generische Problembeschreibung

- kostenpflichtige Nutzung einer Leistung (alle Kosten > 0)
 - einmalige Kosten K für unbeschränkte Nutzung **oder**
 - Kosten $k < K$ für jede Nutzung
- Anzahl Nutzungen n unbekannt
⇒ **(Wann) Soll man K für unbeschränkte Nutzung zahlen?**

Nomenklatur

- Instanz I durch Anzahl Nutzungen n bestimmt
- $OPT(I)$, $ALG(I) \triangleq$ Gesamtkosten

■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$\blacksquare OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$\blacksquare ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c =$

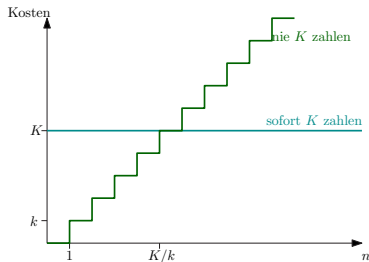
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c =$



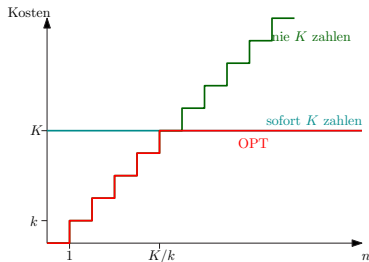
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$\blacksquare OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$\blacksquare ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c =$



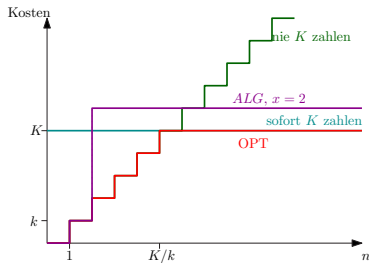
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$\blacksquare OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$\blacksquare ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c =$



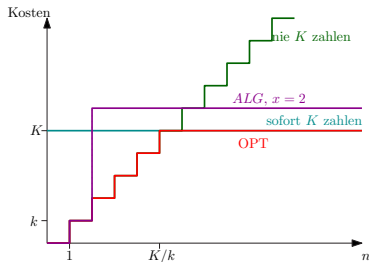
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c = ?$



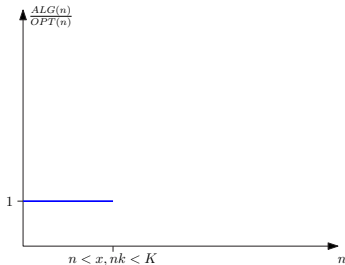
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c = ?$



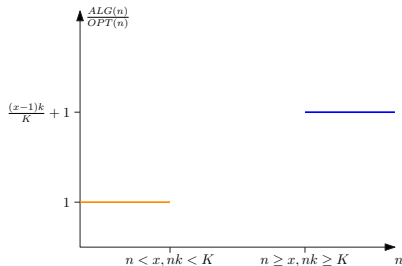
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c = ?$



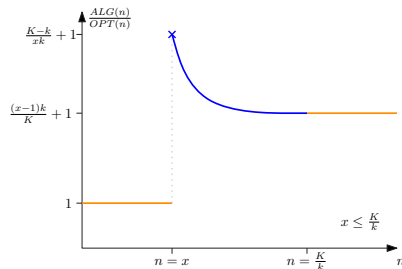
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$\blacksquare OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$\blacksquare ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c = ?$



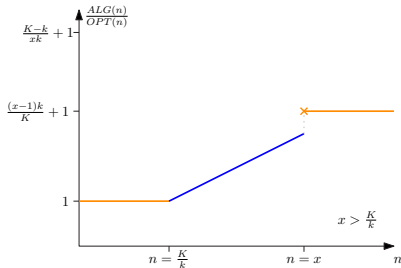
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$\Rightarrow c = ?$

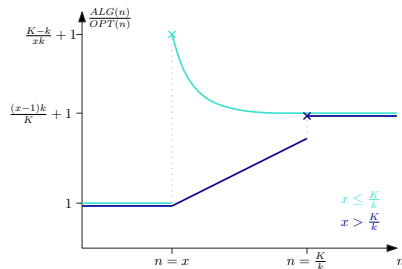


■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

■ $OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$

■ $ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$ (zahle K vor x -ter Nutzung)

$$\Rightarrow c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$

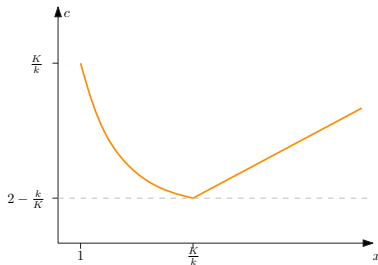


■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

■ $OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$

■ $ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$
(zahle K vor x -ter Nutzung)

$$\Rightarrow c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$



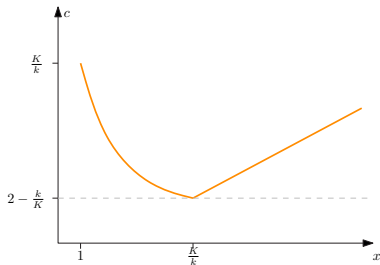
■ allgemein gilt $c = \sup_n \frac{ALG(n)}{OPT(n)}$

$$■ OPT(n) = \begin{cases} n \cdot k & n \cdot k < K \\ K & \text{sonst} \end{cases}$$

$$■ ALG(n) = \begin{cases} n \cdot k & n < x \\ (x-1) \cdot k + K & \text{sonst} \end{cases}$$

(zahle K vor x -ter Nutzung)

$$\Rightarrow c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$



Sonderfall: Bezahle K , nutze nie!

$$\Rightarrow ALG(n) = K$$

$$\Rightarrow c = \infty$$

(entspricht $x = 0$)

$$\text{Es gilt } c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$

Frage: Für welches x ist c minimal?

$$\Rightarrow \text{für } x = \frac{K}{k} \text{ ist } c \text{ minimal mit } c = 2 - \frac{k}{K}$$

Zahlenbeispiel: $K = 100, k = 10 \Rightarrow c = 1.9$

Bester randomisierter Algorithmus: $c = e/(e-1) \approx 1.58$

$$\text{Es gilt } c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$

Frage: Für welches x ist c minimal?

$$\Rightarrow \text{für } x = \frac{K}{k} \text{ ist } c \text{ minimal mit } c = 2 - \frac{k}{K}$$

Zahlenbeispiel: $K = 100, k = 10 \Rightarrow c = 1.9$

Bester randomisierter Algorithmus: $c = e/(e-1) \approx 1.58$

$$\text{Es gilt } c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$

Frage: Für welches x ist c minimal?

$$\Rightarrow \text{für } x = \frac{K}{k} \text{ ist } c \text{ minimal mit } c = 2 - \frac{k}{K}$$

Zahlenbeispiel: $K = 100, k = 10 \Rightarrow c = 1.9$

Bester randomisierter Algorithmus: $c = e/(e-1) \approx 1.58$

$$\text{Es gilt } c = \begin{cases} \frac{K-k}{xk} + 1 & x \leq \frac{K}{k} \\ \frac{(x-1)k}{K} + 1 & \text{sonst} \end{cases}$$

Frage: Für welches x ist c minimal?

$$\Rightarrow \text{für } x = \frac{K}{k} \text{ ist } c \text{ minimal mit } c = 2 - \frac{k}{K}$$

Zahlenbeispiel: $K = 100, k = 10 \Rightarrow c = 1.9$

Bester randomisierter Algorithmus: $c = e/(e-1) \approx 1.58$

Doubling

- Strategie für Online/Offline Approximationsalgorithmen

Idee

- schätze Lösung konservativ ab (eher zu kleine Schätzung)
- prüfe, ob Problem gelöst
- falls nicht erfolgreich, vergrößere Schätzung geometrisch (z.B. verdoppeln, verdreifachen, ...)

Problembeschreibung

- Unbekannter Zielwert $U > 1$
- Bieter gibt eine Reihe an Geboten (b_i) ab bis $b_k \geq U$
- Bieter muss Summe der Gebote $\sum_{i=0}^k b_i$ bezahlen
⇒ Wie kann der Bieter möglichst geschickt vorgehen?

Nomenklatur

- Lösung durch Reihe (b_i) beschrieben
- $OPT(I)$, $ALG(I) \triangleq$ summierten Kosten

Optimaler Offline Algorithmus

- kennt den Zielwert U
- folglich bietet er direkt $b_0 = U$
 $\Rightarrow OPT = U$

Kompetitiver Faktor

$$c = \sup_{k, U} \left\{ \frac{b_0 + b_1 + \dots + b_k}{U} : b_{k-1} < U \leq b_k \right\}$$

- Strategien sind z.B. (a) $b_i = i \cdot x$, (b) $b_i = a \cdot x^i$

Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

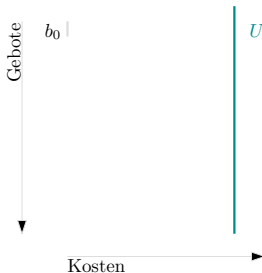
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

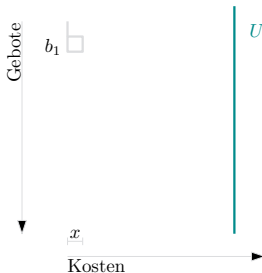
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

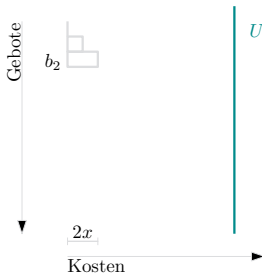
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

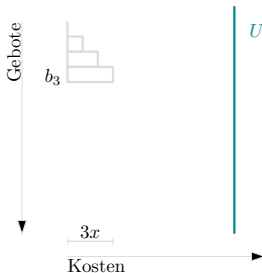
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

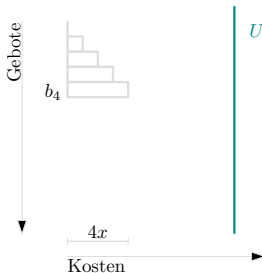
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

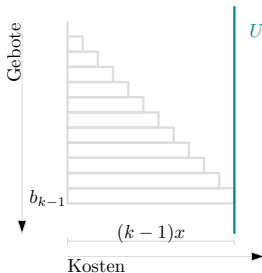
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow C = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

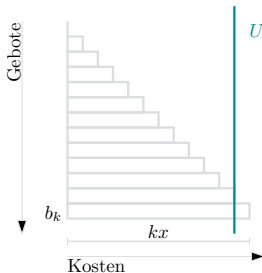
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

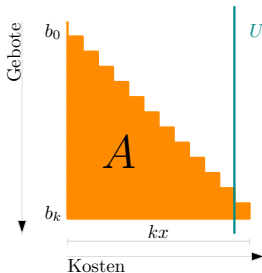
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow C = \infty$$



Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

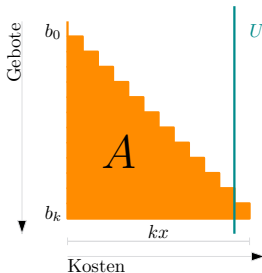
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow C = \infty$$



$$\text{Fläche } A \approx \frac{1}{2} kx \cdot (k+1)$$

Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

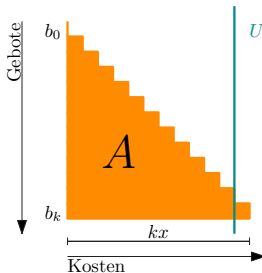
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty$$



$$\text{Fläche } A \approx \frac{1}{2} kx \cdot (k+1)$$

Analyse: (a) $b_i = i \cdot x$ ($x > 0$)

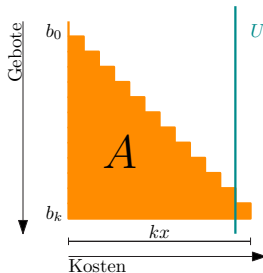
■ **worst case:** $b_{k-1} = (k-1) \cdot x = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k+1 = \left(\frac{U-\varepsilon}{x} + 1 \right) + 1 \leq \frac{U}{x} + 2$$

$$\Rightarrow \text{ALG} = \sum_{i=0}^k i \cdot x = \frac{1}{2}(k+1)k \cdot x$$

$$\Rightarrow \frac{\text{ALG}}{\text{OPT}} \leq \frac{1}{2U} \left(\frac{U}{x} + 2 \right) \left(\frac{U}{x} + 1 \right) \cdot x$$

$$\Rightarrow c = \infty \quad (\text{unabhängig von } x \text{ schlecht})$$



$$\text{Fläche } A \approx \frac{1}{2} kx \cdot (k+1)$$

Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

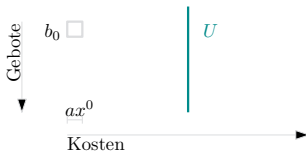
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x - 1} - \frac{a}{(x - 1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x - 1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

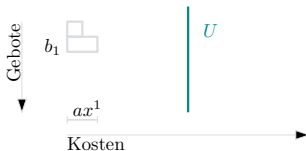
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x - 1} - \frac{a}{(x - 1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x - 1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

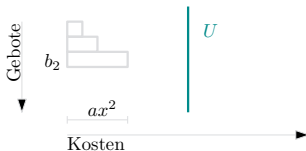
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x - 1} - \frac{a}{(x - 1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x - 1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

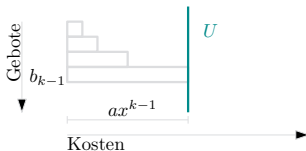
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

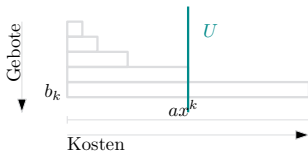
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

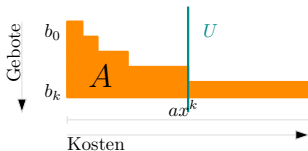
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1}$$



Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

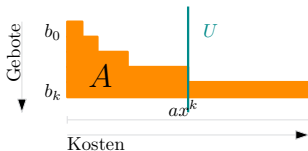
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1}$$



$$\text{Fläche } A \approx \frac{a \cdot (x^{k+1} - 1)}{x - 1}$$

Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

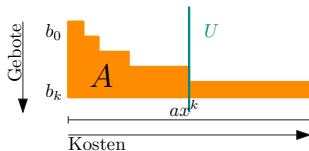
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1}$$



$$\text{Fläche } A \approx \frac{a \cdot (x^{k+1} - 1)}{x - 1}$$

Analyse: (b) $b_i = a \cdot x^i$ ($x > 1$)

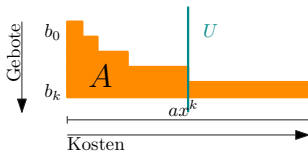
■ *worst case:* $b_{k-1} = a \cdot x^{k-1} = U - \varepsilon$ ($k > 0, \varepsilon \rightarrow 0$)

$$\Rightarrow \text{Anzahl Gebote: } k + 1 = \left(\log_x \left(\frac{U - \varepsilon}{a} \right) + 1 \right) + 1 \leq \log_x \frac{U}{a} + 2$$

$$\Rightarrow ALG = \sum_{i=0}^k a \cdot x^i = \frac{a \cdot (x^{k+1} - 1)}{x - 1} \leq \frac{a \cdot (x^{\log_x \frac{U}{a} + 2} - 1)}{x - 1} = \frac{a \cdot (\frac{U}{a} x^2 - 1)}{x - 1}$$

$$\Rightarrow \frac{ALG}{OPT} \leq \frac{x^2}{x-1} - \frac{a}{(x-1)U}$$

$$\Rightarrow c \leq \frac{x^2}{x-1} \quad (\text{minimal für } x = 2 \Rightarrow c \leq 4)$$



$$\text{Fläche } A \approx \frac{a \cdot (x^{k+1} - 1)}{x - 1}$$

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für gro\ss e } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$s_k = \sum_{i=0}^k b_i$$

$$y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k}$$

$$< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\begin{aligned} \Rightarrow y_k &< (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k} \\ &< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2}) \end{aligned}$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

$$\blacksquare s_k = \sum_{i=0}^k b_i$$

$$\blacksquare y_k = \frac{s_{k+1}}{s_k}$$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\begin{aligned} \Rightarrow y_k &< (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k} \\ &< \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2}) \end{aligned}$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Beweis, dass $c = 4$ untere Schranke ist

■ Definiere:

■ $s_k = \sum_{i=0}^k b_i$

■ $y_k = \frac{s_{k+1}}{s_k}$

■ Annahme: es existiert $c < 4$ mit $\frac{s_k}{U_k} < c$ f.a. k , U_k ($U_k = b_{k-1} + \epsilon_k$)

$$\Rightarrow s_{k+1} < c \cdot (b_k + \epsilon_{k+1}) = c(s_k - s_{k-1} + \epsilon_{k+1})$$

$$\Rightarrow y_k < c(1 - \frac{1}{y_{k-1}} + \frac{\epsilon_{k+1}}{s_k}) < \frac{c}{4} \cdot y_{k-1} + \frac{c \cdot \epsilon_{k+1}}{s_k} \quad (\cdot \frac{1}{s_k}, \text{ mit } 1 - \frac{1}{x} < \frac{x}{4})$$

$$\Rightarrow y_k < \frac{c}{4} \cdot y_{k-1} + 2^{-k} \quad (\text{mit } \epsilon_{k+1} = \frac{s_k}{c \cdot 2^k})$$

$$\Rightarrow y_k < (\frac{c}{4})^1 \cdot y_{k-1} + 2^{-k} < (\frac{c}{4})^2 \cdot y_{k-2} + \frac{c}{4} \cdot 2^{-(k-1)} + 2^{-k} \\ < \dots < (\frac{c}{4})^{k-2} \cdot y_2 + 2^{-3} + 2^{-4} + \dots + 2^{-k} < (\frac{c}{4})^{k-2} \cdot y_2 + (1 - 2^{-2})$$

$$\Rightarrow s_{k+1} < ((\frac{c}{4})^{k-2} \cdot \frac{s_3}{s_2} + (1 - 2^{-2})) \cdot s_k < s_k \quad (\text{für große } k)$$

■ Widerspruch! Annahme falsch.

Wissenswert (nicht erschöpfend!)

- kompetitiver Faktor c
- *Doubling* Technik
- *Ski Rental* Problem

Typische Fragestellungen

- Gegeben sei Algorithmus X . Wie groß ist sein kompetitiver Faktor?
- Geben Sie einen Algorithmus mit kompetitivem Faktor c an.
- Bestimme untere Schranke für kompetitiven Faktor c von Problem P .

Ende!



Feierabend!