DSAL

2. Globalübung

Benjamin Kaminski, Tim Quatmann

4. Mai 2018

Wir sind im Grünen Hörsaal

Agenda

1. $\mathcal{O}$-Notation
2. Average-Case Analyse
3. Rekursionsgleichungen, -bäume, und die Substitutionsmethode

$\mathcal{O}$-Notation

$\mathcal{O}$-Notation

```
foo(int E[], bool b) {
    if (b) {
        bar1(E);
    } else {
        bar2(E)
    }
}
```

- Sei $f(n)$ die Worst-case Laufzeit von `bar1`
- Sei $g(n)$ die Worst-case Laufzeit von `bar2`

- Worst-case Laufzeit von `foo` ? $\max(f(n), g(n))$

$\mathcal{O}$-Notation

Zeige: $\max(f(n), g(n)) \in \Theta(f(n) + g(n))$

$\mathcal{O}$-Notation

$$\begin{aligned}
\text{Zeige:} \quad & \max(f(n), g(n)) \in \Theta(f(n) + g(n)) \\
\Longleftrightarrow \quad & \max(f(n), g(n)) \in \Omega(f(n) + g(n)) \\
\text{und} \quad & \max(f(n), g(n)) \in \mathcal{O}(f(n) + g(n))
\end{aligned}$$

$\max(f(n), g(n)) \in \Omega(f(n) + g(n))$

Zu Zeigen: $\exists c > 0, n_0 \in \mathbb{N}: \forall n \geq n_0:$

$$c \cdot (f(n) + g(n)) \leq \max(f(n), g(n))$$

Es gilt $f(n) + g(n) \leq 2 \cdot \max(f(n), g(n))$

$$\Leftrightarrow \frac{1}{2} \cdot (f(n) + g(n)) \leq \max(f(n), g(n))$$

$\Rightarrow$ Die Obige Ungleichung ist also für $c = \frac{1}{2}$ erfüllt

$\square$

$$\max(f(n), g(n)) \in O(f(n) + g(n))$$

zu Zeigen: $\exists c > 0, n_0 \in \mathbb{N} : \forall n \geq n_0 :$

$$\max(f(n), g(n)) \leq c \cdot (f(n) + g(n))$$

Es gilt $\max(f(n), g(n)) \leq 1 \cdot (f(n) + g(n))$

Die obige Ungleichung gilt also für $c = 1$

$\square$

Average-Case Analyse

Average-Case Analyse

```
foo(bool E[]) {
    int i = 0, m = 1;
    while (i != E.length) {
        if (E[i] == true) { m = 2 * m; }
        i = i+1;
    }
    while (m != 0) { m = m - 1; }
}
```

- Gesucht: $A(n)$
- Alle Eingaben gleich wahrscheinlich
- Nur Vergleiche Kosten Zeit

Beobachtungen:

- Obere Schleife wird immer n mal durchlaufen
  - ⇝ Schleifenbed.: $n+1$ mal überprüft
  - → if-Bed: n mal überprüft
- Untere Schleifenbed: $m+1$ mal überprüft
- $m = 2^k$, wobei $k =$ "Anzahl wahre Einträge"

$$A(n) = n+1 + n + 1 + \sum_{k=0}^{n} 2^k \cdot Pr(\text{"genau } k \text{ wahr"})$$

Sei $\mathcal{E}_k = \{ E \mid E$ hat genau k wahre Einträge$\}$

$$\Pr(\text{"genau } k \text{ Einträge wahr"})$$

$$= \sum_{E \in \mathcal{E}_k} \Pr(E) = \sum_{E \in \mathcal{E}_k} \frac{1}{2^n} = |\mathcal{E}_k| \cdot \frac{1}{2^n}$$

$$= \binom{n}{k} \cdot \frac{1}{2^n}$$

$$\sum_{k=0}^{n} \binom{n}{k} \cdot \frac{1}{2^n} \cdot 2^k = \frac{1}{2^n} \underbrace{\sum_{k=0}^{n} \binom{n}{k} \cdot 2^k}_{(1+2)^n} = \frac{1}{2^n} \cdot 3^n = \left(\frac{3}{2}\right)^n$$

$$A(n) = n+1 + n + 1 + \left(\frac{3}{2}\right)^n = \underline{\underline{2n+2 + \left(\frac{3}{2}\right)^n}}$$

```
foo(bool E[]) {
    int i = 0, m = 1;
    while (i != E.length) {
        if (E[i] == true) { m = 2 * m; }
        i = i+1;
    }
    while (m != 0) { m = m - 1; }
}
```

- Gesucht: $A(n)$
- Alle Eingaben gleich wahrscheinlich
- Nur Vergleiche Kosten Zeit

```
foo(bool E[]) {
    int i = 0, m = 1;
    while (i != E.length) {
        if (E[i] == true) { m = 2 * m; }
        i = i+1;
    }
    while (m != 0) { m = m - 1; }
}
```

- Gesucht: $A(n)$
- Alle Eingaben gleich wahrscheinlich
- Nur Vergleiche Kosten Zeit

Rekursionsgleichungen, -bäume, und die Substitutionsmethode

```
def f(d, l):
    if d > 0:
        return f(d - 1, l/2) + f(d - 1, l/2)
            + f(d - 1, l/2)
    else:
        return (l**2 - (l/2)**2)**(1/2) * l/2
```

8

8

$$F(0, \ell) = 8 \qquad d \leq 0, \quad d = 0, \quad d \in \mathbb{N}$$

$$F(d, \ell) = 3 \cdot F(d-1, \ell/2) + 8 \qquad d > 0$$

Rechenregeln für Logarithmen und Potenzen

Logarithmus:

$b^{?} = y \qquad 1 \cdot \underbrace{b \cdot b \cdot b \cdot \ldots \cdot b}_{? \text{ mal}} \overset{!}{=} y$

$? = \log_b(y)$

- $\boxed{a^{\log_b(c)} = c^{\log_b(a)}}$
- $\left(a^b\right)^c = a^{b \cdot c}$
- $\log_b(a) + \log_b(c) = \log_b(a \cdot c)$
- $\log_b(a) - \log_b(c) = \log_b\left(\frac{a}{c}\right)$
- $\log_b(a) \cdot c = \log_b\left(a^c\right)$

$b^{\log_b(y)} = y$

$a^{\log_b(c)} = \left(b^{\log_b(a)}\right)^{\log_b(c)} = b^{\log_b(a) \cdot \log_b(c)}$

$= \left(b^{\log_b(c)}\right)^{\log_b(a)} = c^{\log_b(a)}$

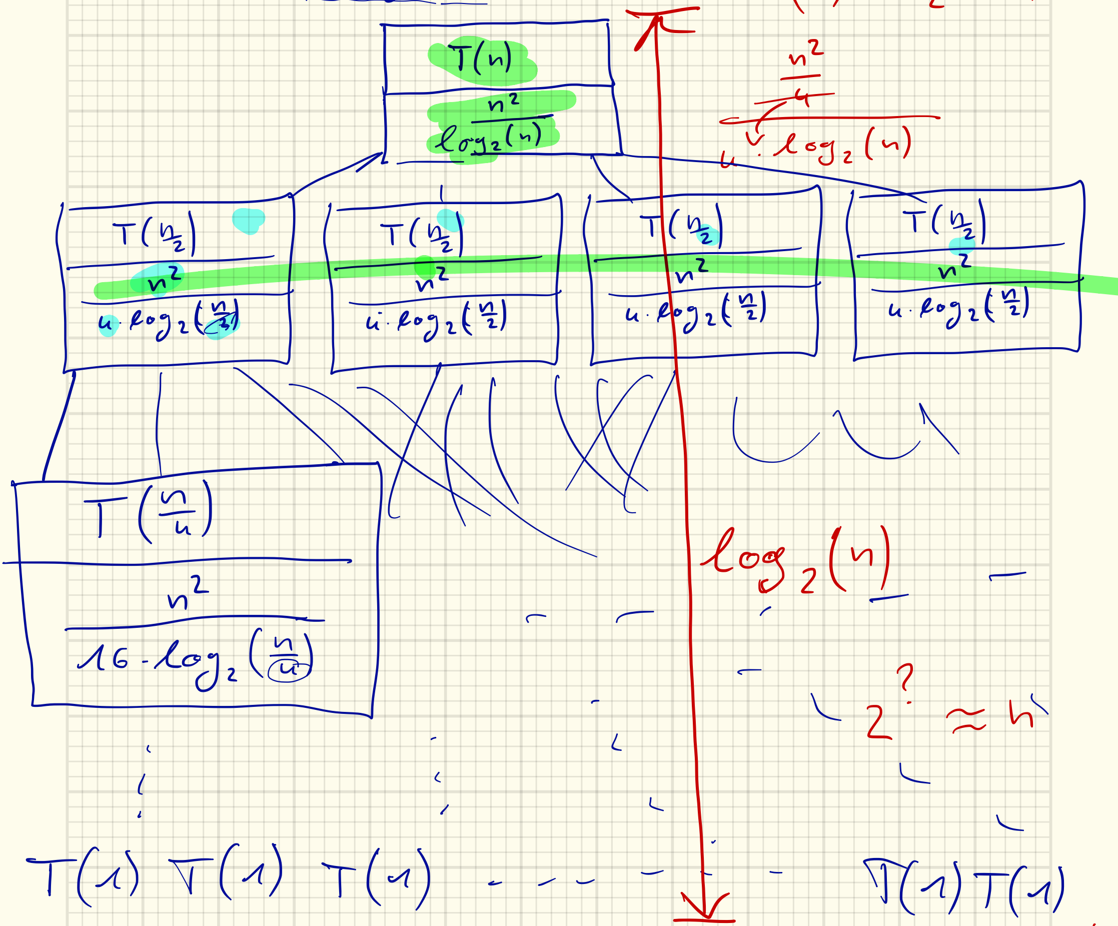
Rekursionsbaum

$$
\begin{aligned}
T(1) &= 1, \\
T(n) &= 4 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \frac{n^2}{\log_2(n)}
\end{aligned}
$$

$$T(1) = 1$$

$$T(n) = 4T\left(\frac{n}{2}\right) + \frac{n^2}{\log_2(n)}$$

$$\left(\frac{n}{2}\right)^2 = \frac{n^2}{2^2} = \frac{n^2}{4}$$



$$4^{\log_2(n)} = n^{\log_2(4)} = n^2$$

$$\sum_{i=0}^{\log_2(n)-1} \left(\frac{n^2}{\log_2\left(\frac{n}{2^i}\right) \cdot \cancel{4^i}} \cdot \cancel{4^i} \right) + n^2 \; \cancel{\in \Theta(n^2)} \in O\left(n^2 \cdot \ln\left(\log_2(n)\right)\right)$$

Substitutionsmethode

$$T(1) = 1$$

$$T(n) = 4 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \frac{n^2}{\log_2(n)}$$

$$T(n) \in \underline{\mathcal{O}\Big(n^2 \cdot \ln(\log_2(n))\Big)}$$

z.z.: $\exists c > 0, n_0 \ \forall n \geq n_0 : T(n) \leq c \cdot n^2 \cdot \ln(\log_2(n))$

IA. $n = 4$ | $T(4) = 4 \cdot T\left(\left\lfloor \frac{4}{2} \right\rfloor\right) + \frac{4^2}{\log_2(4)} = \dots =$

$c \cdot 4^2 \cdot \ln(\log_2(4))$

$\approx 77.6 > 40$ ✓

$= 4 \cdot T(2) + \dots$

$= 4 \cdot \left(4 \cdot T\left(\left\lfloor \frac{2}{2} \right\rfloor\right) + \frac{2^2}{\log_2(2)}\right) + \dots$

$= \dots = 40$

$\underline{n=5}$ $T(5) \approx 42.8 > 40$

$$T\left(\left\lfloor \tfrac{5}{2} \right\rfloor\right) = T\left(\left\lfloor \tfrac{4}{2} \right\rfloor\right)$$

$$\frac{4^2}{\log_2(4)} \neq \frac{5^2}{\log_2(5)}$$

$\boxed{n=7 \text{ auch!!!}}$

$\boxed{n=6 \text{ auch!}}$

I.H. $\forall m < n$: $T(m) \le \underline{c \cdot m^2} \cdot \ln(\log_2(m))$

I.S. $T(n) = 4\, T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \boxed{\frac{n^2}{\log_2(n)}}$

$n > 5$

$$\overset{\text{I.H.}}{\le} 4 \cdot c \left\lfloor \frac{n}{2} \right\rfloor^2 \cdot \ln\left(\log_2\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right) + \frac{n^2}{\log_2(n)}$$

$$\le \cancel{4} \cdot c\, \frac{n^2}{\cancel{4}} \cdot \ln\left(\log_2\left(\frac{n}{2}\right)\right) + \frac{n^2}{\log_2(n)}$$

$\ln(k-1) < \ln(k) - \frac{1}{k}$

$$= c n^2 \cdot \ln\left(\log_2(n) - \log_2(2)\right) + \frac{\cdots}{\cdots}$$

$$= c n^2 \cdot \ln(\log_2(n) - 1) + \frac{n^2}{\log_2(n)}$$

$$< c n^2 \left(\ln(\log_2(n)) - \frac{1}{\log_2(n)}\right) + \frac{n^2}{\log_2(n)}$$

$$= c n^2 \left(\ln(\log_2(n))\right) \underbrace{- c\, \frac{n^2}{\log_2(n)} + \frac{n^2}{\log_2(n)}}_{<0}$$

$$\le c n^2 \ln(\log_2(n)) \checkmark$$

Nächster Termin

Nächste Vorlesung

Montag 7. Mai, 8:30 (H01).

Nächste Globalübung

Dienstag 8. Mai, 13:15 (Aula 1).