Lecture 8: Wednesday January 22, 2003

today

- Chapter 4: recurrences
  - more iterated substitution
  - recursion trees

our story so far

- D&C algorithm often recursive
- analysis of recursive algorithm $\Rightarrow$ recurrence relation
- example: running time of QZ(n)? (say uniform RAM) ...suppose
  - arithmetic, assignment — 1 cycle
  - return, branch test — 1 cycle
  - procedure call — 1 cycle/parameter

```
proc QZ(n)                              time T(n) in cycles
  if n > 1 then                               2
    a <- n * n + 97                           3
    b <- a * QZ(n/2)                          4 + T(n/2)
    return QZ(n/2) * QZ(n/2) + n              7 + T(n/2) + T(n/2)
  else
    return n*n                                2
```

- claim: running time of `QZ(n)` $\in \Theta(t(n))$
- $t(n) = \begin{cases} c_1\,[4] & \text{if} \quad n \leq 1 \\ c_2\,[16] \;+\; 3\,T(\lfloor n/2 \rfloor) & \text{if} \quad n \geq 2 \end{cases}$
- solving $T(n)$?

solving $T(n)$ for $n = 2^k$

- repeated substitution
- see pattern
- guess answer
- prove answer by induction

$$
\begin{aligned}
T(n=2^k) &= 3\,T(\frac{2^k}{2}) + c_2 && \text{def'n of } T(n)\\
&= 3\,T(2^{k-1}) + c_2 && \text{arith.}\\
&= 3\,(3\,T(\frac{2^{k-1}}{2}) + c_2) + c_2 && \text{def'n of } T(n)\\
&= 3^2\,T(2^{k-2}) + 3\cdot c_2 + c_2 && \text{arith.}\\
&= 3^2\,(3\,T(\frac{2^{k-2}}{2}) + c_2) + 3\cdot c_2 + c_2 && \text{def'n of } T(n)\\
&= 3^3\,T(2^{k-3}) + 3^2\cdot c_2 + 3\cdot c_2 + c_2 && \text{arith.}\\
&= 3^3\,(3\,T(\frac{2^{k-3}}{2}) + c_2) + 3^2\cdot c_2 + 3\cdot c_2 + c_2 && \text{def'n of } T(n)\\
&= 3^4\,T(2^{k-4}) + (3^3 + 3^2 + 3^1 + 3^0)c_2 && \text{arith.}\\
&\ldots\\
&= 3^k\,(T(2^0)) + (3^{k-1} + \ldots + 3^1 + 3^0)c_2 && \text{guess pattern}\\
&= 3^k c_1 + \frac{3^k-1}{3-1}\,c_2 && \text{geometric sum}\\
&= (c_1 + \frac{c_2}{2})3^k - \frac{c_2}{2} && \text{arith.}
\end{aligned}
$$

proof by induction that

$$T(n = 2^k) = (c_1 + \frac{c_2}{2})3^k \; - \; \frac{c_2}{2} \qquad \forall k \geq 0$$

base case

$$\begin{aligned} T(n = 2^0) \; &= \; c_1 && \text{by def'n} \\ &= \; (c_1 + \frac{c_2}{2})3^0 \; - \; \frac{c_2}{2} \; = \; c_1 && \text{so claim holds in this case} \end{aligned}$$

inductive case

assume claim holds for $k = t - 1 \geq 0$

want to show claim holds for $k = t$

$$\begin{aligned} T(n = 2^t) \; &= \; 3\,T(n/2) \; + \; c_2 && \text{def'n} \\ &= \; 3\,T(2^{t-1}) \; + \; c_2 && \text{arith.} \\ &= \; 3\,(\,(c_1 + \frac{c_2}{2})\,3^{t-1} \; - \; \frac{c_2}{2}) \; + \; c_2 && \text{ind. ass'n} \\ &= \; (c_1 + \frac{c_2}{2})\,3^t \; - \; \frac{c_2}{2} && \text{arith.} \end{aligned}$$

so claim holds in this case, so claim holds for all $k \geq 0$ 

summary

- run time analysis $\Rightarrow T(n) = \begin{cases} c_1 & \text{if} \quad n \leq 1 \\ c_2 \ + \ 3\,T(\lfloor n/2 \rfloor) & \text{if} \quad n \geq 2 \end{cases}$
- goal: solve $T(n)$
  - repeated subst'n
  - guess
  - proof
- $$T(n = 2^k) \ = \ (c_1 + \frac{c_2}{2})\,3^k \ - \ \frac{c_2}{2}$$
$$\in \ \Theta(3^k) \ = \ \Theta(3^{\lg n}) \ = \ \Theta(n^{\lg 3})$$
- can show: $T(n) \ \in \ \Theta(n^{\lg 3})$ for all $n \geq 1$

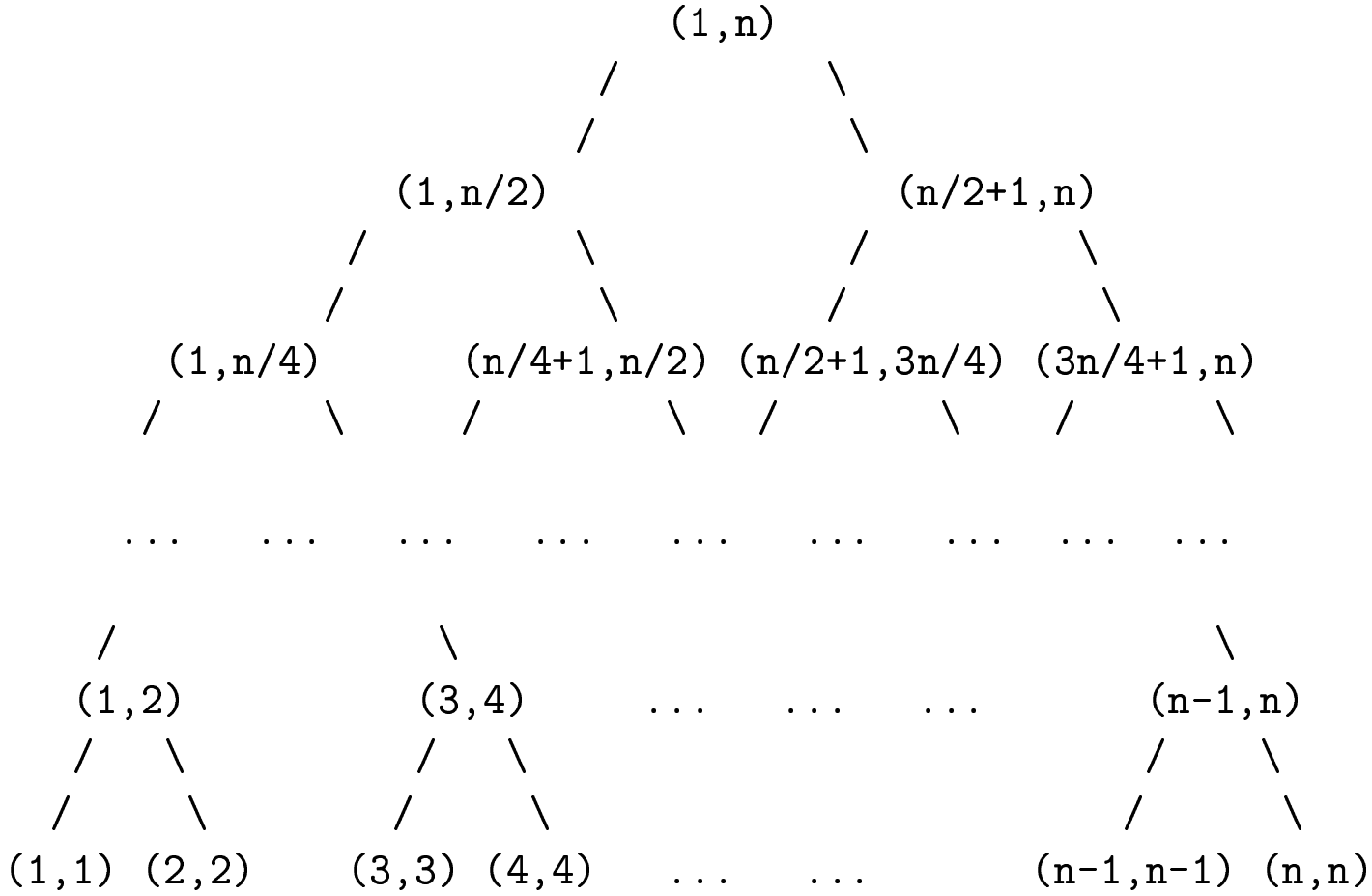
mergesort analysis

- merge sorted lists, sizes x and y
  - WC $x + y - 1$ key comparisons
  - BC $\min\{x, y\}$ key comparisons
- $W(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ W(\lfloor n/2 \rfloor) + W(\lceil n/2 \rceil) + n - 1 & \text{if } n \geq 2 \end{cases}$
- $W(n) \leq T(n)$, where

  $T(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n & \text{if } n \geq 2 \end{cases}$
- solve for $T(n = 2^k)$

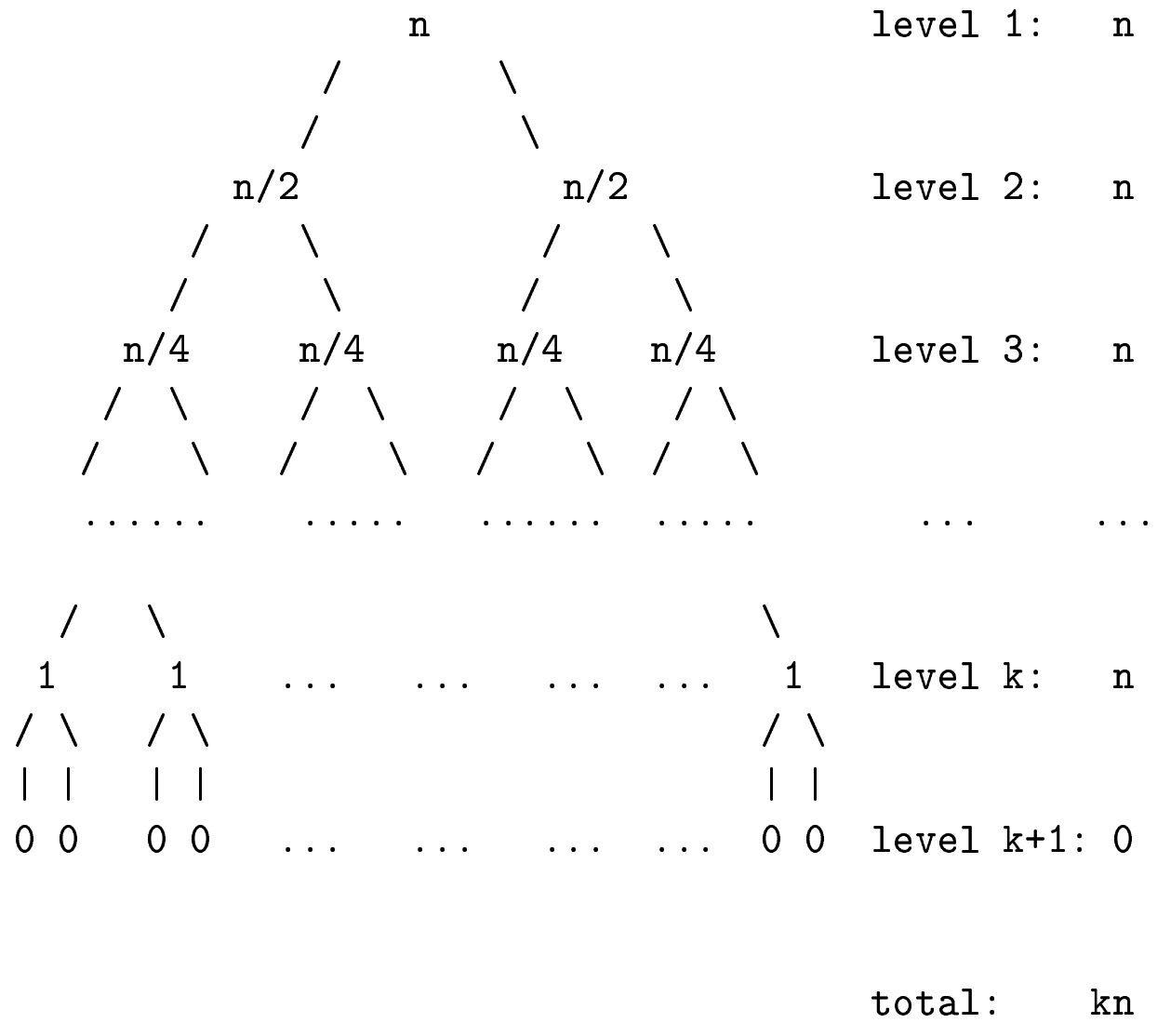
$$
\begin{aligned}
T(n=2^k) &= 2\,T(2^k/2) + 2^k && \text{def'n of T} \\
&= 2\,T(2^{k-1}) + 2^k && \text{arith.} \\
&= 2\,[2\,T(2^{k-1}/2) + 2^{k-1}\,] + 2^k && \text{def'n} \\
&= 2^2\,T(2^{k-2}) + 2^k + 2^k && \text{arith.} \\
&= 2^2\,[2\,T(2^{k-2}/2) + 2^{k-2}\,] + 2^k + 2^k + 2^k && \text{def'n} \\
&= 2^3\,T(2^{k-3}) + 2^k + 2^k + 2^k && \text{arith.} \\
&= 2^3\,[2\,T(2^{k-3}/2) + 2^{k-3}\,] + 2^k + 2^k + 2^k && \text{def'n} \\
&= 2^4\,T(2^{k-4}) + 4\,2^k && \text{arith.} \\
&\quad \ldots \\
&= 2^k\,T(2^0) + k\,2^k && \text{guess} \\
&= k\,2^k && \text{defn} \\
&= n \lg n && \text{since } n = 2^k
\end{aligned}
$$

mergesort recursion tree (showing parameters of each call)



mergesort recursion tree (showing KC per call)

- assuming merge(n) takes $\approx n$ KC, how many KC per level?



solving recurrence relations

- iterated substitution (done)
- recursion trees (done)
- master theorem (now)
- what form do D&C r.r.'s usually have?
- consider the following D&C procedure:

```
proc yada(n)
    ...  ...  ...  ...  ...  ...  ...
    ... yada(n/b)  ...  yada(n/b) ...
    ...  ...  ...  ...  ...  ...  ...
    ...  ...   return   ...
endYada
```

- for the call `yada(n)` assume:
  - run time (excluding recursive calls) is n^c
  - there are a total of a calls to `yada(n/b)`
- recurrence relation for total time $T(n)$:

$$T(n) = \begin{cases} aT(\frac{n}{b}) + n^c & \text{if } n \geq b \\ \text{bounded} & \text{if } n < b \end{cases}$$

- closed form solution?
  - repeated substitution
  - simplifying assumption: $n = b^k$