Lecture 9: Monday Sept 24, 2001

today

- Chapter 4: recurrences
  - iterated substitution
  - recursion trees
  - master theorem

announcements

- assignment: name, id, acknowledgement (else no mark)
- assignment 3 due next Monday in class
- midterm: next Wednesday in class
- turkey trot this Saturday

recall from last lecture ...

- $T(n) = \begin{cases} 1 & \text{if} \quad n \le 1 \\ 3\,T(\lfloor n/2 \rfloor) \; + \; 5 & \text{if} \quad n \ge 2 \end{cases}$

- solving $T(n)$ for $n = 2^k$
  - repeated substitution
  - see pattern
  - guess answer
  - prove answer by induction

$$
\begin{aligned}
T(n = 2^k) &= 3\,T(2^{k-1}) \; + 5 && \text{subst'n} \\
&= 3^2\,T(2^{k-2}) \; + 3 \cdot 5 \; + 5 && \text{subst'n} \\
&= 3^3\,T(2^{k-3}) \; + 3^2 \cdot 5 \; + 3 \cdot 5 \; + 5 && \text{subst'n} \\
&= 3^4\,T(2^{k-4}) \; + \; (3^3 \; + \; 3^2 \; + \; 3^1 \; + \; 3^0)5 && \text{subst'n} \\
&= \ldots \\
&= 3^k\,(T(2^0)) \; + \; (3^{k-1} \; + \; \ldots \; + \; 3^1 \; + \; 3^0)5 && \text{guess pattern} \\
&= 3^k \; + \; \frac{3^k - 1}{3 - 1}\,5 && \text{geometric sum} \\
&= \frac{7}{2}\,3^k \; - \; \frac{5}{2} && \text{arith.} \\
&= \frac{7}{2}\,3^{\lg n} \; - \; \frac{5}{2} && \text{arith.}
\end{aligned}
$$

proof by induction that

$$T(n = 2^k) = \frac{7}{2}3^k \; - \; \frac{5}{2} \qquad \forall k \geq 0$$

base case

$$\begin{aligned} T(n = 2^0) \;&=\; 1 && \text{by def'n} \\ &=\; \frac{7}{2}3^0 \; - \; \frac{5}{2} && \text{so claim holds in this case} \end{aligned}$$

inductive case

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

want to show claim holds for $k = t$

$$\begin{aligned} T(n = 2^t) \;&=\; 3\,T(n/2) \; + \; 5 && \text{def'n} \\ &=\; 3\,T(2^{t-1}) \; + \; 5 && \text{arith.} \\ &=\; 3\,(\frac{7}{2}3^{t-1} \; - \; \frac{5}{2}) \; + \; 5 && \text{ind. ass'n} \\ &=\; \frac{7}{2}3^t \; - \; \frac{5}{2} && \text{arith.} \end{aligned}$$

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

summary

- run time analysis $\Rightarrow T(n) = \begin{cases} 1 & \text{if} \quad n \leq 1 \\ 3\,T(\lfloor n/2 \rfloor) \, + \, 5 & \text{if} \quad n \geq 2 \end{cases}$
- goal: solve $T(n)$
  - repeated subst'n
  - guess
  - proof
-

$$T(n = 2^k) \; = \; \frac{7}{2}3^k \; - \; \frac{5}{2}$$

$$\in \; \Theta(3^k) \; = \; \Theta(3^{\lg n}) \; = \; \Theta(n^{\lg 3})$$

- can show: $T(n) \; \in \; \Theta(n^{\lg 3}) \quad$ 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 $T(n = 2^k)$

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

mergesort recursion tree (showing parameters of each call)

```
                             (1,n)
                          /         \
                         /           \
               (1,n/2)                  (n/2+1,n)
             /         \              /          \
            /           \            /            \
    (1,n/4)        (n/4+1,n/2) (n/2+1,3n/4) (3n/4+1,n)
   /        \     /         \   /        \    /      \

  ...    ...    ...    ...    ...    ...    ...   ...   ...

  /              \                                      \
 (1,2)          (3,4)      ...    ...    ...        (n-1,n)
  /   \          /   \                                /    \
 /     \        /     \                              /      \
(1,1) (2,2)    (3,3) (4,4)    ...    ...        (n-1,n-1) (n,n)
```

mergesort recursion tree (showing KC per call)

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

```
                  n                   level 1:   n
              /       \
             /         \
          n/2            n/2          level 2:   n
        /    \          /    \
       /      \        /      \
     n/4     n/4      n/4    n/4      level 3:   n
    /  \     /  \     /  \    / \
   /    \   /    \   /    \  /   \
   ......    .....   ......  .....       ...     ...

  /   \                           \
 1     1    ...   ...   ...  ...   1  level k:   n
/ \   / \                         / \
| |   | |                         | |
0 0   0 0   ...   ...   ...  ...  0 0 level k+1: 0


                                      total:     kn
```

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: $n = b^k$

$$
\begin{aligned}
T(n) &= aT(\frac{n}{b}) + n^c \\
&= a[aT(\frac{n/b}{b}) + (n/b)^c] + n^c \\
&= a^2T(\frac{n}{b^2}) + a(\frac{n}{b})^c + n^c \\
&= a^2[aT(\frac{n/b^2}{b}) + (n/b^2)^c] + a(\frac{n}{b})^c + n^c \\
&= a^3T(\frac{n}{b^3}) + a^2(\frac{n}{b^2})^c + a(\frac{n}{b})^c + n^c \\
&= \ldots \\
&= a^kT(\frac{n}{b^k}) + a^{k-1}(\frac{n}{b^{k-1}})^c + a^{k-2}(\frac{n}{b^{k-2}})^c + \ldots + a(\frac{n}{b})^c + n^c \\
&= T(1)\, a^{\lg_b n} + n^c \left[\sum_{j=0}^{k-1} \left(\frac{a}{b^c}\right)^j \right] \\
&=^* T(1)\, n^{\lg_b a} + \begin{cases} n^c \lg_b n & \text{if } \ a = b^c \\ n^c \dfrac{1 - \left(\dfrac{a}{b^c}\right)^{\lg_b n}}{1 - \dfrac{a}{b^c}} & \text{if } \ a \neq b^c \end{cases}
\end{aligned}
$$

* Recall that $\displaystyle\sum_{j=0}^{k-1} \alpha^j = \begin{cases} k & \text{if } \ \alpha = 1 \\ \dfrac{1-\alpha^k}{1-\alpha} & \text{if } \ \alpha \neq 1 \end{cases}$

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

- if $a = b^c$ then $\lg_b a = c$ and

$$T(n) = T(1)\, n^{\lg_b a} + n^c \lg_b n \quad \in \Theta(n^c \lg n)$$

- if $a < b^c$ then $\lg_b a < c$ and

$$T(n) = T(1)\, n^{\lg_b a} + n^c \frac{1 - (a/b^c)^{\lg_b n}}{1 - a/b^c} \quad \in \Theta(n^c)$$

- if $a > b^c$ then $\lg_b a > c$ and

$$\begin{aligned} T(n) &= T(1)\, n^{\lg_b a} + n^c \frac{(a/b^c)^{\lg_b n} - 1}{(a/b^c) - 1} \\ &\in \Theta(\ n^{\lg_b a} + n^c \left(\frac{a}{b^c}\right)^{\lg_b n}\ ) \\ &\in \Theta(\ n^{\lg_b a} + n^c \frac{a^{\lg_b n}}{(b^c)^{\lg_b n}}\ ) \\ &\in \Theta(\ n^{\lg_b a} + n^c \frac{n^{\lg_b a}}{n^c}\ ) \\ &\in \Theta(\ n^{\lg_b a}\ ) \end{aligned}$$

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

- if $a = b^c$ then $\lg_b a = c$ and $\qquad T(n) \in \Theta(n^c \lg n)$
- if $a < b^c$ then $\lg_b a < c$ and $\qquad T(n) \in \Theta(n^c)$
- if $a > b^c$ then $\lg_b a > c$ and $\qquad T(n) \in \Theta(n^{\lg_b a})$

- same result if n^c replaced by $f(n) \in \Theta(n^c)$

- algorithm design consequences:
  - $\lg_b a = c$? $\qquad$ balance: $T(n) \uparrow$ if $\lg_b a$ or $c \uparrow$
    redesign: $\qquad$ no
  - $\lg_b a < c$? $\qquad$ rec'n slack: $T(n) \downarrow$ if $c \downarrow$ $\lg_b a \uparrow$
    redesign: $\qquad$ less body, more/bigger calls
  - $\lg_b a > c$? $\qquad$ body slack: $T(n) \downarrow$ if $\lg_b a \downarrow$ $c \uparrow$
    redesign: $\qquad$ fewer/smaller calls, more body

☺

some examples ...

$$T(n) = \begin{cases} 2T(n/2) + n & \text{if } n \geq 2 \\ T(1) & \text{if } n = 1 \end{cases}$$

$a = 2$, $b = 2$, $c = 1$, $a = b^c$ so $T(n) \in \Theta(n \lg n)$

$$T(n) = \begin{cases} 4T(n/2) + n^3 & \text{if } n \geq 2 \\ T(1) & \text{if } n = 1 \end{cases}$$

$a = 4$, $b = 2$, $c = 3$, $a < b^c$ so $T(n) \in \Theta(n^3)$

$$T(n) = \begin{cases} 7T(n/2) + n^2 & \text{if } n \geq 2 \\ T(1) & \text{if } n = 1 \end{cases}$$

$a = 7$, $b = 2$, $c = 2$, $a > b^c$ so $T(n) \in \Theta(n^{\lg_2 7})$

$$T(n) = \begin{cases} 7T(n/2) + n^2/\lg n & \text{if } n \geq 2 \\ T(1) & \text{if } n = 1 \end{cases}$$

$a = 7$, $b = 2$, $c =$??? ...so formula does not apply

exercise: in the last example, show that $T(n) \in O(n^{\lg_2 7})$