Lecture 7: Recurrences

Agenda:

- Iterated substitution — more examples
- Recursion trees — the second method

Reading:

- Textbook pages 62 – 72

Solving the recurrence:(from last lecture)

- $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3\texttt{QZ}(\frac{n}{2}) + 5 & \text{if } n \geq 2 \end{cases}$

- Again, <u>Iterated Substitution</u>

$$
\begin{aligned}
& T(2^k) \\
= \;& 3 \times T(2^{k-1}) + 5 \\
= \;& 3 \times \left(3 \times T(2^{k-2}) + 5\right) + 5 \\
= \;& 3^2 \times T(2^{k-2}) + 3 \times 5 + 5 \\
= \;& 3^2 \times \left(3 \times T(2^{k-3}) + 5\right) + 3 \times 5 + 5 \\
= \;& 3^3 \times T(2^{k-3}) + 3^2 \times 5 + 3 \times 5 + 5 \\
= \;& \ldots \\
= \;& 3^k \times T(2^{k-k}) + 3^{k-1} \times 5 + 3^{k-2} \times 5 + \ldots + 3 \times 5 + 5 \\
= \;& 3^k + 5 \times \left(\sum_{i=0}^{k-1} 3^i\right) \\
= \;& 3^k + 5 \times \left(\frac{3^k - 1}{2}\right) \\
= \;& 3.5 \times 3^k - 2.5
\end{aligned}
$$

- So, $T(n) = 3.5 \times 3^{\lg n} - 2.5 = 3.5 \times n^{\lg 3} - 2.5$.

- Remember: prove by induction later.

The exercise:

- Examine the running time of $\texttt{QZ}(n)$ (uniform cost RAM)

```
Proc QZ(n)

if n > 1 then
    a ← n × n + 37
    b ← a × QZ(n/2)
    return QZ(n/2) × QZ(n/2) + n
else
    return n × n
```

- Uniform cost RAM:
  - Arithmetic, assignment — 1 cycle
  - Return, branch test — 1 cycle
  - Procedure call — 1 cycle

-

Proc $\texttt{QZ}(n)$	time $T(n)$ in cycles
if $n > 1$ **then**	2
$\quad a \leftarrow n \times n + 37$	3
$\quad b \leftarrow a \times \texttt{QZ}(\frac{n}{2})$	$4 + T(\frac{n}{2})$
$\quad$ **return** $\texttt{QZ}(\frac{n}{2}) \times \texttt{QZ}(\frac{n}{2}) + n$	$7 + T(\frac{n}{2}) + T(\frac{n}{2})$
else	
$\quad$ **return** $n \times n$	2

$\texttt{QZ}(n)$ running time:

- Claim: running time of $\texttt{QZ}(n) \in \Theta(T(n))$
- $$T(n) = \begin{cases} 4, & \text{if } n = 1 \\ 3 \times T(\frac{n}{2}) + 16, & \text{if } n \geq 2 \end{cases}$$
- Solving $T(n)$?

  Solving $T(n)$ for $n = 2^k$ — iterated substitution

  - Substitution enough steps to *see the pattern*
  - Guess the pattern and prove by induction

$$\begin{array}{rll}
 & T(n = 2^k) & \\
= & 3 \times T(\frac{2^k}{2}) + 16 & \text{definition} \\
= & 3 \times T(2^{k-1}) + 16 & \text{arithmetic} \\
= & 3 \times \left(3 \times T(\frac{2^{k-1}}{2}) + 16\right) + 16 & \text{definition} \\
= & 3^2 \times T(2^{k-2}) + 3 \times 16 + 16 & \text{arithmetic} \\
= & 3^2 \times \left(3 \times T(\frac{2^{k-2}}{2}) + 16\right) + 3 \times 16 + 16 & \text{definition} \\
= & 3^3 \times T(2^{k-3}) + 3^2 \times 16 + 3 \times 16 + 16 & \text{arithmetic} \\
= & 3^3 \times \left(3 \times T(\frac{2^{k-3}}{2}) + 16\right) + 3^2 \times 16 + 3 \times 16 + 16 & \text{definition} \\
= & 3^4 \times T(2^{k-4}) + 3^3 \times 16 + 3^2 \times 16 + 3 \times 16 + 16 & \text{arithmetic} \\
\ldots & & \\
= & 3^k \times T(2^{k-k}) + 3^{k-1} \times 16 + 3^{k-2} \times 16 + & \\
 & \quad \ldots + 3^2 \times 16 + 3 \times 16 + 16 & \text{guess} \\
= & 3^k \times 4 + 16 \times \dfrac{3^k - 1}{3 - 1} & \\
= & 12 \times 3^k - 8 &
\end{array}$$

- So, next to prove $T(2^k) = 12 \times 3^k - 8$, for $k \geq 0$

Proof by induction $T(2^k) = 12 \times 3^k - 8$:

- Base step: $k = 0$

  According to guessed $T(2^0) = 12 - 8 = 4$.

  So, it holds for base case.

- Inductive step:

  Assume that $T(2^{k-1}) = 12 \times 3^{k-1} - 8$.

  By recurrence relation

  $$T(2^k) = 3 \times T(\frac{2^k}{2}) + 16 = 3 \times T(2^{k-1}) + 16,$$

  so

  $T(2^k) = 3 \times \Big(12 \times 3^{k-1} - 8\Big) + 16 = 12 \times 3^k - 24 + 16 = 12 \times 3^k - 8.$

  Thus, it holds for inductive step too.

- Therefore, $T(2^k) = 12 \times 3^k - 8$ holds for any $k \geq 0$.

Summary:

- Running time analysis of `QZ`$(n) \Longrightarrow$
  analysis of function
  $$T(n) = \begin{cases} c_1, & \text{if } n = 1 \\ 3 \times T(\frac{n}{2}) + c_2, & \text{if } n \geq 2 \end{cases}$$

- Solving $T(n)$
  1. Iterated (Repeated) substitution
  2. Guess the pattern and get the closed form
  3. Proof by induction

- $T(n = 2^k) = (c_1 + \frac{c_2}{2})3^k - \frac{c_2}{2} \in \Theta(3^k) = \Theta(3^{\lg n}) = \Theta(n^{\lg 3})$.

- Can show: $T(n) \in \Theta(n^{\lg 3})$ for all $n \geq 1$
  — dealing with floor/ceiling

Merge sort analysis:

- Recurrence (last lecture):

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ (n-1) + 2 \times T(\frac{n}{2}) & \text{if } n \geq 2 \end{cases}$$

- Guessed closed form and proved by induction (last lecture):

$$T(n) = n(\lg n - 1) + 1, n \geq 1$$

- Look at the partition tree:

$$(1, n)$$

$$(1, \tfrac{n}{2}) \qquad (\tfrac{n}{2}+1, n)$$

$$(1, \tfrac{n}{4}) \quad (\tfrac{n}{4}+1, \tfrac{n}{2}) \; (\tfrac{n}{2}+1, \tfrac{3n}{4}) \quad (\tfrac{3n}{4}+1, n)$$

$$\cdots\cdots\cdots$$

$$(1,2) \qquad (3,4) \qquad \cdots\cdots \qquad (n-1, n)$$

$$(1,1)\;(2,2) \quad (3,3)\;(4,4) \;\cdots\; (n-1, n-1)\;(n,n)$$

- Question: the number of KC per cell?

Merge sort recursion tree (KC per cell):

Assuming `merge`(n) takes $\sim n$ KC:

$(1, n)$ — `level 1:` n

$(1, \frac{n}{2})$ $(\frac{n}{2} + 1, n)$ — `level 2:` n

$(1, \frac{n}{4})$ $(\frac{n}{4} + 1, \frac{n}{2})$ $(\frac{n}{2} + 1, \frac{3n}{4})$ $(\frac{3n}{4} + 1, n)$ — `level 3:` n

$\cdots\cdots\cdots$ — $\cdots$

$(1, 2)$ $(3, 4)$ $\cdots\cdots$ $(n-1, n)$ — `level` k`:` n

$(1, 1)$ $(2, 2)$ $(3, 3)$ $(4, 4)$ $\cdots$ $(n-1, n-1)$ (n, n) — `level` $k+1$`:` 0

`total:` kn

Where $k = \lg n$!!!

Solving recurrence relations:

- Iterated substitution (done)
- Recursion tree (done)
- Master theorem (next)
- Divide-and-conquer: what form of recurrence relation does it have?
- Typical procedure:

```
Proc dnq(n)

    ......
    dnq(n/b) ... dnq(n/b)
    ......
    return
end dnq
```

- For the call dnq(n) assume:
  - running time (excluding recursive calls) is n^c
  - there are a total of a calls to dnq($\frac{n}{b}$)
- Recurrence relation for total time $T(n)$

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

- Closed form solution?
  - Iterated substitution !!!
  - Simplifying assumption to $n = b^k$

Have you understood the lecture contents?

well	ok	not-at-all	topic
☐	☐	☐	iterated substitution method
☐	☐	☐	closed form guessing
☐	☐	☐	prove by math induction
☐	☐	☐	recursion tree
☐	☐	☐	operations per cell in the tree
☐	☐	☐	general divide-and-conquer recurrence