Agenda:

- Iterated substitution more examples
- Recursion trees the second method

Reading:

• Textbook pages 62 – 72

Solving the recurrence:(from last lecture)

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

• Again, Iterated Substitution

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

- 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 QZ(n) (uniform cost RAM)

```
Proc QZ(n)

if n > 1 then

a \leftarrow n \times n + 37

b \leftarrow a \times QZ(\frac{n}{2})

return QZ(\frac{n}{2}) \times QZ(\frac{n}{2}) + n

else

return n \times n
```

#### • Uniform cost RAM:

- Arithmetic, assignment 1 cycle
- Return, branch test
   1 cycle
- Procedure call

• Proc QZ(n) time T(n) in cycles

 $\begin{array}{ll} \text{if } n > 1 \text{ then} & 2 \\ a \leftarrow n \times n + 37 & 3 \\ b \leftarrow a \times \mathbb{QZ}(\frac{n}{2}) & 4 + T(\frac{n}{2}) \\ \text{return } \mathbb{QZ}(\frac{n}{2}) \times \mathbb{QZ}(\frac{n}{2}) + n & 7 + T(\frac{n}{2}) + T(\frac{n}{2}) \\ \text{else} & 2 \end{array}$ 

3

1 cycle

QZ(n) running time:

- Claim: running time of  $QZ(n) \in \Theta(T(n))$
- •

$$T(n) = \begin{cases} 4, & \text{if } n = 1\\ 3 \times T(\frac{n}{2}) + 16, & \text{if } n \ge 2 \end{cases}$$

- Solving T(n)?
   Solving T(n) for n = 2<sup>k</sup> iterated substitution
  - Substitution enough steps to see the pattern
  - Guess the pattern and prove by induction

$$\begin{array}{rcl} T(n=2^k) & = & 3 \times T(\frac{2^k}{2}) + 16 & & \text{definition} \\ = & 3 \times T(2^{k-1}) + 16 & & \text{arithmetic} \\ = & 3 \times \left(3 \times T(\frac{2^{k-1}}{2}) + 16\right) + 16 & & \text{definition} \\ = & 3^2 \times T(2^{k-2}) + 3 \times 16 + 16 & & \text{arithmetic} \\ = & 3^2 \times \left(3 \times T(\frac{2^{k-2}}{2}) + 16\right) + 3 \times 16 + 16 & & \text{definition} \\ = & 3^3 \times T(2^{k-3}) + 3^2 \times 16 + 3 \times 16 + 16 & & \text{arithmetic} \\ = & 3^3 \times \left(3 \times T(\frac{2^{k-3}}{2}) + 16\right) + 3^2 \times 16 + 3 \times 16 + 16 & & \text{arithmetic} \\ = & 3^4 \times T(2^{k-4}) + 3^3 \times 16 + 3^2 \times 16 + 3 \times 16 + 16 & & \text{arithmetic} \\ \end{array}$$

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

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

- Base step: k = 0
   According to guessed T(2<sup>0</sup>) = 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

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

Thus, it holds for inductive step too.

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

Summary:

Running time analysis of QZ(n) ⇒
 analysis of function

$$T(n) = \begin{cases} c_1, & \text{if } n = 1\\ 3 \times T(\frac{n}{2}) + c_2, & \text{if } n \ge 2 \end{cases}$$

- 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  $n \ge 1$ — dealing with floor/ceiling

Merge sort analysis:

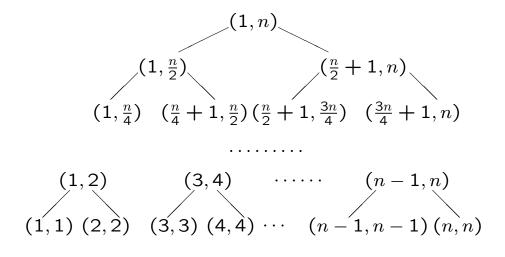
• Recurrence (last lecture):

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

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

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

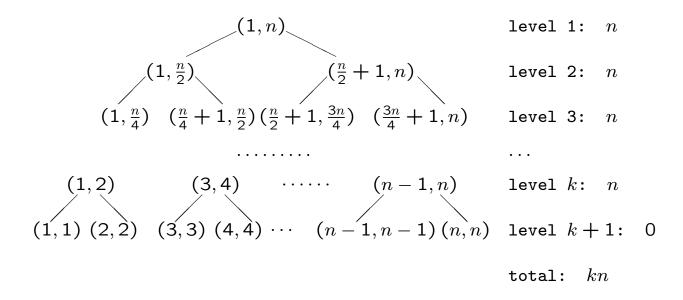
• Look at the partition tree:



• Question: the number of KC per cell?

Merge sort recursion tree (KC per cell):

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



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)

\dots

dnq(\frac{n}{b}) \dots dnq(\frac{n}{b})

\dots

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)

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

- 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