

6 Sept. 2023

What does it take to prove an algo is correct?

① Termination: The algorithm terminates

(a) give asymptotic analysis ~~??~~

(b) decrementing fn  $\leftarrow$  we'll get to this

T

② Partial Correctness: If the algorithm terminates, then it is correct.

$\rightarrow$  usu. through loop / recursion invariant (we'll get to this)

$T \Rightarrow C$

$\therefore C$

(modus ponens!)

Find (x, A)

Runtime

$n = |A|$

$\Theta(1)$  1:  $i \leftarrow 1$

2: while  $i \leq |A|$

$\Theta(1)$  3: | if  $x = A[i]$

$\Theta(1)$  4: | return TRUE

$\Theta(1)$  5: | endif

$\Theta(1)$  6: |  $i++$

$\Theta(1)$  7: end while

$\Theta(1)$  8: return FALSE

Runtime is

$\underbrace{\Theta(1)}_{\text{before loop}} + n \cdot \underbrace{\Theta(1)}_{\text{loop}} + \underbrace{\Theta(1)}_{\text{after the loop}}$

$= \Theta(n) + 2\Theta(1)$

$= \boxed{\Theta(n)}$

# Recursive Runtimes

① Binary Search an array of size  $n \rightarrow$  runtime  $T(n)$

- Runtime:
- 1: check middle  $\Theta(1)$
  - 2: recursively call on half  $T(n/2)$
  - 3: Use recursive ans as my own  $\Theta(1)$

$$T(n) = T(n/2) + \Theta(1)$$

Let's be a bit specific for now.

Suppose steps 1 & 3 take  $d$  units of time.  
 $\uparrow$  a constant

Suppose  $T(1) = c$ .

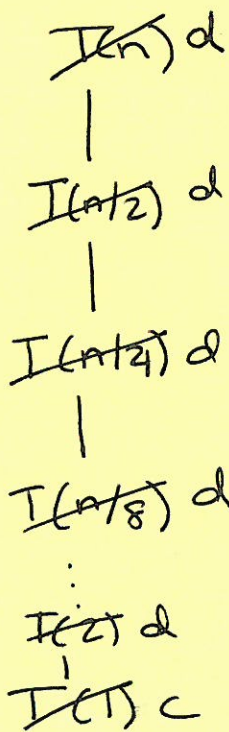
$$T(n) = \begin{cases} c, & n=1 \\ T(n/2) + d, & n>1 \end{cases}$$

both are Recursive Form

note: when using  $\Theta$ 's in recursive form, can ignore base case (it's always  $\Theta(1)$ !)

Recursion Tree:

(Assume  $n$  is a power of 2)



how many levels?  
 $\log_2 n + 1$ ,  
all of which except the last "cost" me  $d$ .  
 $\therefore$  total RT/cost is

$$d \cdot \log_2 n + c$$

$$T(n) = d \log_2 n + c$$

closed form  
"exact formula"

Asymptotic Form:  
 $T(n) \in \Theta(\log n)$

## ② Recurrence Relation:

$$T(n) = \begin{cases} 1, & n = 1 \\ T(n-1) + n, & n > 1 \end{cases}$$

Questions

a) What is the closed form?  $T(n) = \sum_{i=1}^n i$  ← summation form

b) Asymptotic Form?  $\Theta(n^2)$   $T(n) = \frac{n(n+1)}{2}$

c) Name an algorithm with the recurrence  $T(n) = T(n-1) + \Theta(n)$ .

↳ selection sort

examples

$$n = 3$$

with recurrence relation:

$$\begin{aligned} T(3) &= T(2) + 3 \quad \left. \begin{array}{l} \text{via} \\ \text{recursion} \end{array} \right\} \\ &= (T(1) + 2) + 3 \\ &= (1 + 2) + 3 \quad \left. \begin{array}{l} \text{via} \\ \text{base} \\ \text{case} \end{array} \right\} \\ &= 1 + 2 + 3 = 6 \\ &= \sum_{i=1}^3 i \end{aligned}$$

with closed form:  $T(3) = \frac{3(4)}{2} = 3 \cdot 2 = 6$

with recursion tree

$$\left. \begin{array}{l} \bullet T(3) \ 3 \\ | \\ \bullet T(2) \ 2 \\ | \\ \bullet T(1) \ 1 \end{array} \right\} \begin{array}{l} \text{Sum values on} \\ \text{my nodes} \\ \sum_{i=1}^n i = \sum_{i=1}^n n - (i-1) \end{array}$$

# Solving Recurrence Relations

If we have no better option:

- Guess + Check!
- ↳ gut feeling
  - top-down: unravel via the recursion tree
  - bottom-up: start w/ small #s + see if a pattern is found.
- ↳ induction, either for closed form or asymptotic.