30 August 2021

Announcements:

- volunteer?
- exams: 22 Sept, 9 Nov. (I'll double check), December
- respond to survey if you haven't yet!
- H-00 due tonight: <u>both</u> to D2L + to gradescope.
- office hours: Mondays 9:30-10:30 Zoom
  Wednesdays after class

$\boxed{Q}$ How do you prove that an algorithm is correct?

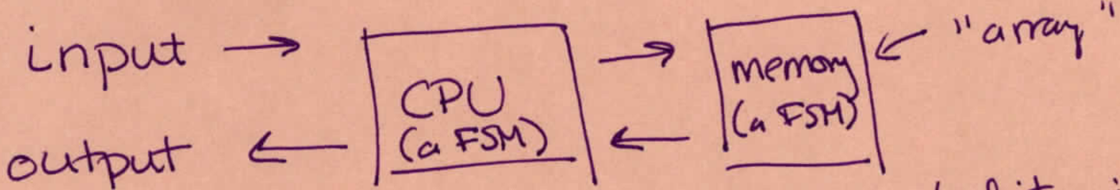
1. Proof of Termination
   - → runtime analysis "This algorithm takes $\Theta(n)$ time"
   - → proving that it does end (might not know how long it takes though)
2. Partial Correctness "If it terminates, then it is correct."
   - → Loop / Recursion Invariant

$$
\begin{array}{l}
\text{(1)}\ T \\
\text{(2)}\ T \Rightarrow C \\
\hline
\therefore C
\end{array}
$$

Model of Computation: Real RAM MOC

"memory units" are real #s.

Random Access Machine (RAM)
is a pair of Finite State Machines.



- +, −, ÷, × in const. time
- other simple ops.

- access/edit in const. time
- not differentiating btween cashed/not cashed
- data is stored, where 1 $\mathbb{R}$-number = 1 unit of memory

"simplified reality"
→ the math is done w/ $\mathbb{R}$, but computations are w/ 1's and 0's.
→ + and × are not equals! (in reality)
→ unlimited memory!

* Know what is constant-time operations in the MOC that you are using!!

contrary to: Turing Machine: one FSM + infinite tape
"linked list"

value on array of values

FIND (x, A)

$i \leftarrow 1$	$1 = \Theta(1)$
while $i \leq \|A\|$	$3 = \Theta(1)$
if $x =$ i^{th} value of A	$3 = \Theta(1)$
return TRUE	$\Theta(1)$
endif noop	$\Theta(1)$
end while	$\Theta(1)$
return FALSE	$\Theta(1)$

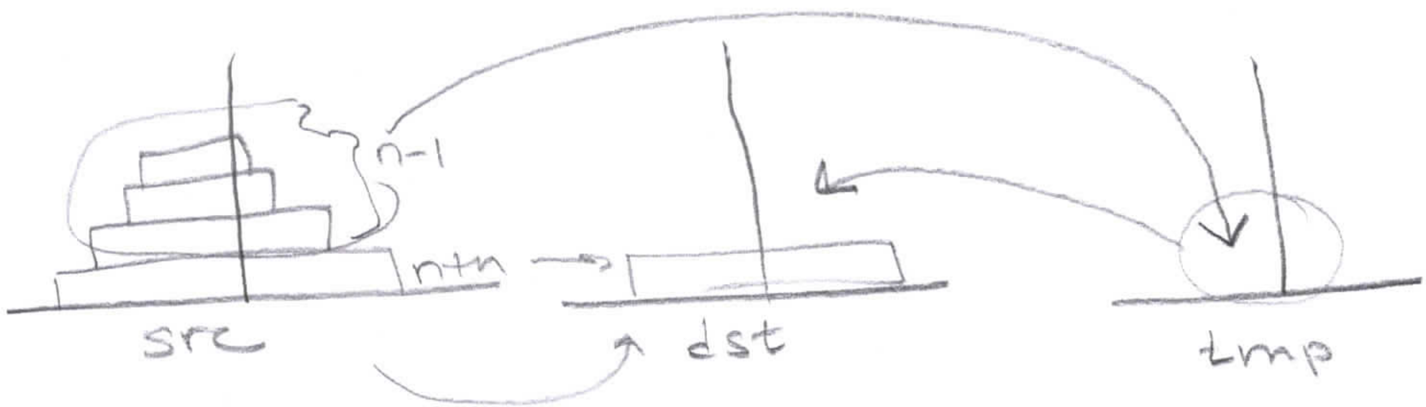
$|A| \times$

linear search to find x!
(used b/c we don't know if A is sorted or not!)

$$\Theta(1) + |A|\left(5\Theta(1)\right) + \Theta(1)$$

$$= |A|(5+2)(\Theta(1))$$

$$= \Theta(|A|)$$




```
HANOI(n, src, dst, tmp):
Θ(1)      if n > 0
T(n-1)        HANOI(n − 1, src, tmp, dst)     ⟨⟨Recurse!⟩⟩
Θ(1)          move disk n from src to dst
T(n-1)        HANOI(n − 1, tmp, dst, src)     ⟨⟨Recurse!⟩⟩
2T(n-1) + Θ(1)
```

Figure 1.4. A recursive algorithm to solve the Tower of Hanoi

Let $T(n)$ denote the runtime of HANOI when we have n disks.

So, $T(n) = 2T(n-1) + \Theta(1)$ recursive formulation

$\boxed{Q}$ How do I simplify this?

$T: \mathbb{N} \to \mathbb{R}$

$$T(n) = \begin{cases} 1, & n = 1 \\ 2T(n-1) + 1, & n > 1 \end{cases}$$ recursive form

$T(n) = 2^n - 1$ ← closed form

$\Theta(2^n)$ ← asymptotic form

$\boxed{\text{OPTION 1}}$: Guess & check
↳ via induction!

$\boxed{\text{OPTION 2}}$ Educated Guess → try for a small #
"bottom up"
↳ draw the recursion tree
"top down"

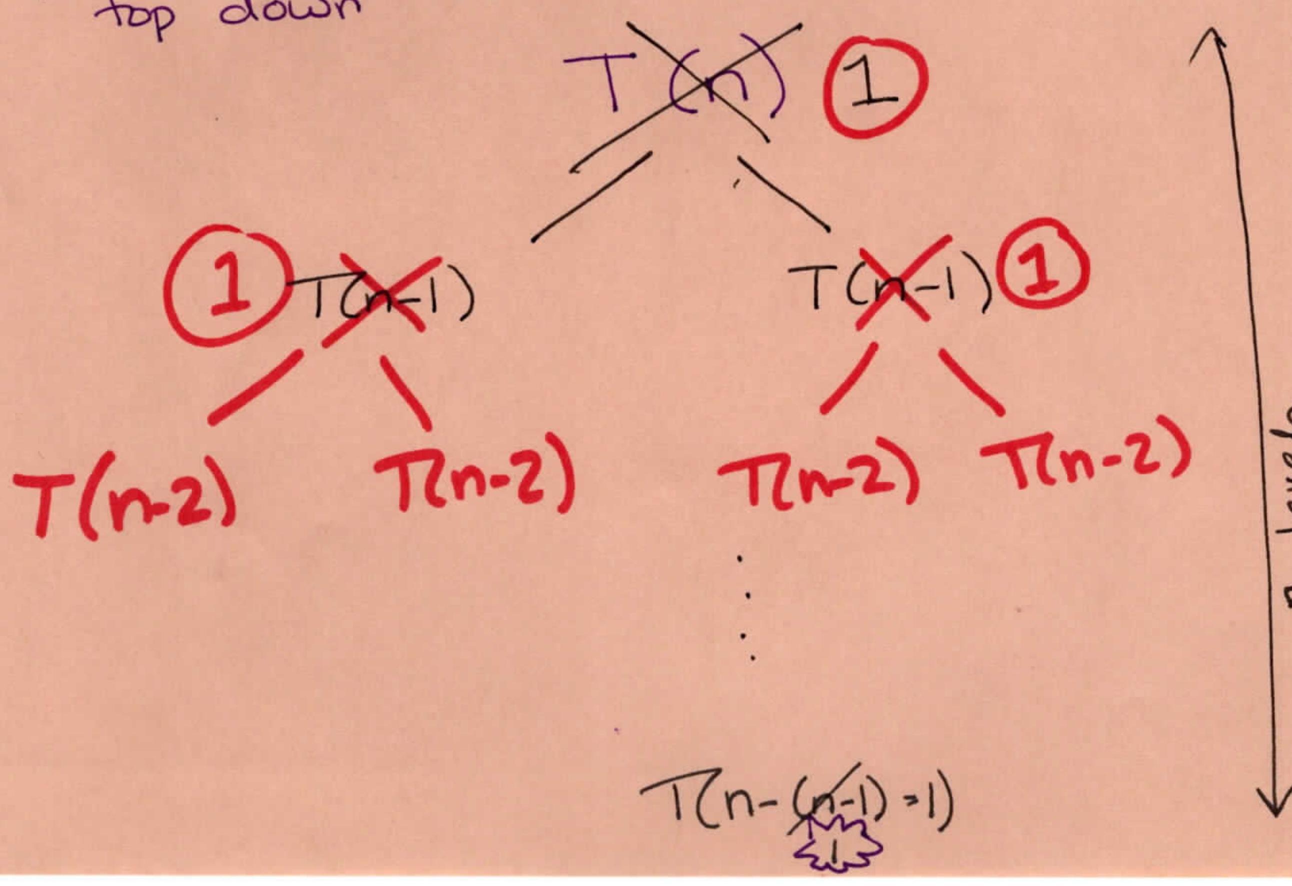
"bottom up"

$T(1) = 1 = 2 - 1$

$T(2) = 2T(1) + 1 = 2 + 1 = 3 = 4 - 1$

$T(3) = 2T(2) + 1 = 2 \cdot 3 + 1 = 7 = 8 - 1$

"top down"



Q: How many nodes are in this tree? $2^n - 1$ + each costs 1