CSC236 Feb 1, 2018

Note: Office hours Friday 12-1 BA 8135

Continued... Recall we left off doing a running time complexity analysis of Merge Sort

$$\begin{aligned}
T(n) &= 2\cdot T(n/2) + bn = 2[2\cdot T(n/4) + b\,n/2] + bn \\
&= 2^2 T(n/2^2) + 2b\,n/2 + bn = 2^2 T(n/2^2) + 2bn \\
&= 2^2\cdot[2\,T(n/2^3) + b\,n/2^2] + 2bn = 2^3 T(n/2^3) + 2^2 b\,n/2^2 + 2bn \\
&= 2^3 T(n/2^3) + 3bn = 2^i = T(n/2^i) + ibn \text{ on ith iteration}
\end{aligned}$$

↳ Prove this is true for all i iterations using induction.

- $n/2^i = 1$ when $2^i = n$ ∴ when $i = \log_2 n$; then $\log_2 n$ iterations bring us to base case.
- $T(n) = n\,T(1) + \log_2 n \cdot b \cdot n = \boxed{a\cdot n + bn\log_2 n}$ //closed form

- Master theorem definition on Slide 18.
15 - $cT(n/d)$: time of recursive call on $1/d$th size problem
- $f(n)$: time to merge results from recursive calls
- whichever one is the dominant factor is what affects performance degradation
- always make sure master theorem applys before using it
  ↳ there will be trick questions on tests that look Master Theoremy

2 - when $f(n) \notin O(n^k)$, we may approximate $f(n)$ with a polynomial.
  ↳ can use squeeze theorem to find bounds.
6 - sometimes Master theorem used to compare different algos

- no amount of testing can replace a proof of correctness

- turing completeness (computable numbers paper) means programs can be written which do not terminate.

- Precondition + Algo → Termination
  Precondition + Algo + termination → Post condition

➢ Proving correctness of recursive functions:

(6) - encapslate partial correctness & Termination in one predicate

- $P(n)$: (pre cond) $\wedge$ (runs) $\wedge$ (input size n) $\rightarrow$ (halts) $\wedge$ (partial correct)

5 • $P(n)$: "If $n \in \mathbb{N}$ and if the program runs and if the size of the input is n, then the program halts & $Sq(n)$ returns n^2"

• Prove by simple induction claim: $\forall n \in \mathbb{N}\ p(n)$

• Base $n=0$: $n \in \mathbb{N}$, runs, input $n \rightarrow$ returns $0 = 0^2 \therefore P(0)$ true

• Assume $P(k)$ true for arbitrary $k \in \mathbb{N}$,

$\therefore P(k+1)$: calls $Sq(k)$ returns (k^2)

returns $[k^2 + 2(k+1) - 1] = k^2 + 2k + 1 = (k+1)^2$

Then $P(k+1)$ terminates and the post-condition holds.

So $P(k) \rightarrow P(k+1)\ \forall k \in \mathbb{N}$

• $P(0) \wedge [P(k) \rightarrow P(k+1)] \therefore \forall n \in \mathbb{N}\ P(n)$ holds.

• Then via PSI can conclude that $Sq(n)$ returns correct answer for all $n \in \mathbb{N}$