CSE 101 – Oct 4, 2019 (Week 1)

Notes provided by Ben Sihota bsihota@ucsc.edu

Course Notes

- PA1 extended till Thursday
- Make a CrowdGrader account
- HW due Monday

Analyzing Recursive Algorithms

To analyze recursive algorithms, write a recurrence

$$\begin{cases} c & 1 \leq n < n_0 \\ aT(\frac{n}{b}) + D(n) + C(n) & n \leq n_0 \end{cases}$$

- $T(n)$ is the worst case number of basics operations on input of size n
- $aT(\frac{n}{b})$ is a sub-instance each of size $\frac{n}{b}$
- $D(n)$ is the cost of dividing
- $C(n)$ is the cost of combining

Goal: Find an asymptotic solution to this recurrence

Discussion of PA1

- Several TAs at each lab
- Don't implement a sorting algorithm
- ADT must be separate from freestanding program
- Okay to call exit if precondition not meant
- Error message include: module, which operation, and which precondition

Handout on asymptotic growth

Definition: Let $f(n)$ and $g(n)$ be functions

$O(g(n)) = \{f(n) | \exists c > o, \exists n_0 > 0, \forall n \geq n_0 : 0 \leq f(n) \leq cg(n)\}$

I.e. $f(n) \in O(g(n))$ iff there exists pos. c, n_0 such that $\forall n \geq n_0$

We say: $f(n)$ is "order" $g(n)$

We write: $f(n) = O(g(n))$

Geometrically: *Graph: $cg(n)$ is above $f(n)$*

Note: c and n_0 are **not** unique

Example $10n + 100 = O(n^2 - 40n + 400)$

Pick $n_0 = 40$ and $c = 1$

In fact:

$$an + b = O(cn^2 + dn + e)$$

for any a, b, c, d, e with $a > 0, c > 0$

Note

If $f(n) = O(g(n))$, then

- $f(n) \geq 0$ for sufficiently large n
- $g(n) \geq 0$ for sufficiently large n

Definition: $f(n)$ is asymptotically non-negative iff $\exists n_0 > 0$ such that $\forall n \geq n_0$: $f(n) \geq 0$

(Asymptotically positive: $f(n) > 0$)

Blanket assymption: All functions under discussion are asymptotically non-negative

In fact, if $P(n)$ and $Q(n)$ are polynomials with

$$deg(P(n)) \leq deg(Q(n))$$

then

$$P(n) = O(Q(n))$$

If $f(n)$ and $g(n)$ are asymptotically positive, then our definition is equivalent to

$$\frac{f(n)}{g(n)} \leq c$$

For some c, n_0 and all $n \geq n_0$

Definition:

$\Omega(g(n)) = \{f(n) | \exists c > o, \exists n_0 > 0, \forall n \geq n_0 : \Theta \leq c : g(n) \geq f(n)\}$

Notation: $f(n) = \Omega(g(n))$

We say:

- if $f(n) = O(g(n)) : g(n)$ is an asymptotic upper bound
- if $f(n) = \Omega(g(n)) : g(n)$ is an asymptotic lower bound