CS 161 (Stanford, Winter 2023) Section 1

Asymptotic Analysis

Asymptotic Analysis Definitions

Let f, g be functions from the positive integers to the non-negative reals. **Definition 1:** (Big-Oh notation)

$f = O(g)$ if there exist constants $c > 0$ and n_0 such that for all $n > n_0$,

$$f(n) \leq c \cdot g(n).$$

Definition 2: (Big-Omega notation)

$f = \Omega(g)$ if there exist constants $c > 0$ and n_0 such that for all $n > n_0$,

$$f(n) \geq c \cdot g(n).$$

Definition 3: (Big-Theta notation)

$f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

Note: You will use "Big-Oh notation", "Big-Omega notation", and "Big-Theta notation" A LOT in class. Additionally, you may occasionally run into "little-oh notation" and "little-omega notation": **Definition 4:**(Little-oh notation)

$f = o(g)$ if **for every constant** $c > 0$ there exist a constant n_0 such that for all $n > n_0$,

$$f(n) < c \cdot g(n).$$

Definition 5:(Little-omega notation)

$f = \omega(g)$ if **for every constant** $c > 0$ there exist a constant n_0 such that for all $n > n_0$,

$$f(n) > c \cdot g(n).$$

1 Asymptotic Analysis Problems

1.1

Prove that if $f = \Omega(g)$ then f is not in $o(g)$.

1.2

For each of the following functions, prove whether $f = O(g)$, $f = \Omega(g)$, or $f = \Theta(g)$. For example, by specifying some explicit constants n_0 and $c > 0$ such that the definition of Big-Oh, Big-Omega, or Big-Theta is satisfied. *Bonus: prove little-Oh and little-Omega*

(a)	$f(n) = n \log(n^3)$	$g(n) = n \log n$
(b)	$f(n) = 2^{2n}$	$g(n) = 3^n$
(c)	$f(n) = \sum_{i=1}^{n} \log i$	$g(n) = n \log n$

1.3

Give an example of f, g such that f is not $O(g)$ and g is not $O(f)$.

2 Matrix Multiplication

In lecture, you have seen how digit multiplication can be improved upon with divide and conquer. Let us see a more generalized example of Matrix multiplication. Assume that we have matrices A and B and we'd like to multiply them. Both matrices have n rows and n columns. *For this question, you can make the simplifying assumption that the product of any two entries from A and B can be calculated in O(1) time.*

2.1

What is the naive solution and what is its runtime? Think about how you multiply matrices.

2.2

Now if we divide up the problem like this:

$$XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

We now have a divide and conquer strategy! Find the recurrence relation of this strategy and the runtime of this algorithm.

2.3

Can we do better? It turns out we can by calculating only 7 of the sub problems:

$$P_1 = A(F - H) \qquad P_5 = (A + D)(E + H)$$
$$P_2 = (A + B)H \qquad P_6 = (B - D)(G + H)$$
$$P_3 = (C + D)E \qquad P_7 = (A - C)(E + F)$$
$$P_4 = D(G - E)$$

And we can solve XY by

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

We now have a more efficient divide and conquer strategy! What is the recurrence relation of this strategy and what is the runtime of this algorithm?

3 How NOT to prove claims by induction

In this class, you will prove a lot of claims, many of them by induction. You might also prove some wrong claims, and catching those mistakes will be an important skill! The following are examples of a false proof where an obviously untrue claim has been 'proven' using induction (with some errors or missing details, of course). Your task is to investigate the 'proofs' and identify the mistakes made.

3.1

Fake Claim 1: For every non-negative integer n, $2^n = 1$. **Inductive Hypothesis:** for all integers n such that $0 \leq n \leq k$, $2^n = 1$. **Base Case:** For $n = 0$, $2^0 = 1$. **Inductive Step:** Suppose the inductive hypothesis holds for k; we will show that it is also true $k+1$, i.e. $2^{k+1} = 1$. We have

$$\begin{aligned} 2^{k+1} &= \frac{2^{2k}}{2^{k-1}} \\ &= \frac{2^k \cdot 2^k}{2^{k-1}} \\ &= \frac{1 \cdot 1}{1} && \text{(by strong induction hypothesis)} \\ &= 1 \end{aligned}$$

Conclusion: By strong induction, the claim follows.

3.2

Fake Claim 2:

$$\underbrace{\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \ldots}_{n \text{ terms}} = \frac{3}{2} - \frac{1}{n}. \tag{1}$$

Inductive Hypothesis: (1) holds for $n = k$ **Base Case:** For $n = 1$,

$$\frac{1}{1 \cdot 2} = 1/2 = \frac{3}{2} - \frac{1}{1}.$$

Inductive Step: Suppose the inductive hypothesis holds for $n = k$; we will show that it is also true $n = k + 1$. We have

$$\begin{aligned}\left(\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\dots\frac{1}{(k-1)\cdot k}\right)+\frac{1}{k\cdot(k+1)} &= \frac{3}{2}-\frac{1}{k}+\frac{1}{k\cdot(k+1)} \quad \text{(by weak induction hypothesis)}\\ &= \frac{3}{2}-\frac{1}{k}+\frac{1}{k}-\frac{1}{k+1}\\ &= \frac{3}{2}-\frac{1}{k+1}.\end{aligned}$$

Conclusion: By weak induction, the claim follows.

4 Induction: Snowball Fight

On a flat ice sheet, an *odd* number of penguins are standing such that their pairwise distances to each other are all different. At the strike of dawn, each penguin throws a snowball at another penguin that closest to them. Show that there is always some penguin that doesn't get hit by a snowball.