Lecture-02-CMSC351

- Quiz #1 is due tonight, homework #1 is due next week

Title:

- To measure the performance of an algorithm, we use the term **running time**.
- The running time is expressed as a function of the number of inputs, allowing us to compare the relative efficiency of multiple algorithms.
- We look at orders of magnitude when comparing running times between algorithms.

Induction:

- Proof process:
    - Check that the formula is correct for base case.
    - Assuming the formula is correct for n, prove that it is correct for n + 1 (inductive step).

Paper Notes On Next Page:

Ex:1:

1. Prove $T(N) = O(f(n))$

$T(N) = 3N^2 + 2N + 1$

$T(N) = 3n^2 + 2N + 1 \leq 3N^2 + 2N + N$ when $n \geq 1$

$T(N) \leq 3n^2 + 3n \leq 3n^2 + 3n^2 = 6n^2$

$T(N) \leq 6n^2$

$T(N) < cf(N)$

$\therefore$ proved $T(N) = O(f(n)) = O(\theta n^2)$ when $c = 6$, $N = 1$

Proof format: $T(n)$ is $O(f(n))$ if and only if there exists positive constants, N_0 and C such that:

i) $T(n) \leq C \cdot f(n)$ for any $n \geq n_0$

Ex:2: $T(N) = 3N^2 - 2N - 1 \leq 3N^2 - 1$ when $N \geq 0$

$\leq 3N^2$

$\therefore T(N) = 3N^2 - 2N - 1 = O(N^2)$ when $C = 3$, $N = 0$

Ex:3: $T(N) = N \cdot \log_2(N) + N - 3 = O(N \cdot \log_2(N))$

~~$N \cdot \log_2(N) + N - 3 \leq N \cdot \log_2(N)$ when $N = 3$, $C = 1$~~

Must find constants $c > 0$ and $N > 0$ such that

$N \cdot \log_2(N) + N - 3 \leq C \cdot N \cdot \log_2(N)$ for all $N \geq N_0$

For $N \geq 1$: $N \cdot \log_2(N) + N - 3 \leq N \log_2(N) + N$

For $N \geq 2$: $N \cdot \log_2(N) + N \leq N \log_2(N) + N \log_2(N)$

Since $N \leq N \cdot \log_2(N)$

$\therefore T(N) = O(N^2)$ for $N \geq 2$, $C = 2$