Efficient

Wednesday, January 15, 2025 11:36 AM

Last Time:
- Algorithms
- Examples:
  - Closest Values
  - Long Multiplication
  - Peasant Multiplication
- Describing Algorithms

Today
- Remarks about multiplication
- RAM Model
- "Efficient"
  - Polynomials
  - Worst case running time
  - Big-O Notation

MikTeX

Recall : Multiplication of two numbers

$X[1 \ldots m], \; Y[1 \ldots n]$

Lattice mult

$\boxed{O(mn)}$

Peasant mult

$O(\log x \log y)$

$\boxed{O(m\,n)}$

1950s Kolmogrov's Conjecture

"There is no algorithm to multiply numbers in subquadratic time"

RAM model → random access machine

· basic data → int, floats, booleans.

Constant word size. → w

~~PRAM~~

integers are w-bits always.

integers are w-bits always.

$x \cdot y \to O(m \cdot n) \longrightarrow O(w^2) \to O(1)$ ✓

* In the RAM model, arithmetic takes constant time.

Efficiency

"Efficient"

① what is "efficient"?

Brute-force is considered "inefficient"!

Easy example

Given a list of L items, put in sorted order!

Brute-force: Try all orderings. One of them is the sorted order.

↳ $n! \longrightarrow > 2^n$

Exponential time

say running time is 2^n ....

growth is very fast.

· We want running time to increase by constant

· We want running time to increase by constant factor as input size grows.

"efficient"

Def polynomial time.

There exists some constant c and d where on input size n, the running time is at most $c \cdot n^d$.

In this class: Try to beat brute-force.

Question: Are there problems which require exp. time?

---

② Worst case analysis.

Given a list L and a target T, my algorithm finds the index of the first occurence of T in L. (or none).

Alg

① Iterate through L and check whether the current item is T.

② Return the index (or none).

Running time

worst-case time → $O(n)$ ← n is $|L|$

↳ pessimistic.

alternatives → best case → unrealistic.

average case → assume some probability on the input, and then do stuff.

(3) Big-O notation

Let $T(n)$ be the running time of an alg.

$T(n) = 5n^2 - 3n + 1$

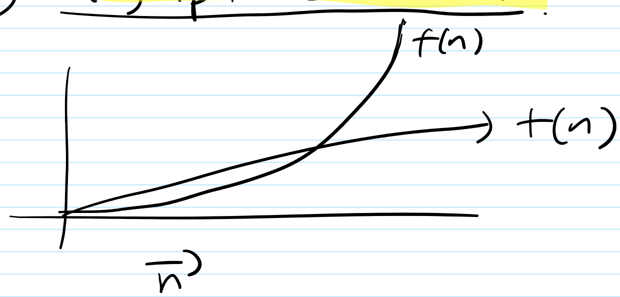
↳ "$T(n)$ is $O(n^2)$"

Def We say $T(n)$ is $O(f(n))$ if there exists $c > 0$ and $n_0 \geq 0$ where:

for all $n \geq n_0$, we have $T(n) \leq c \cdot f(n)$

① upperbound up to constant factor

② asymptotic bound



ex $T(n) = 5n^2 + 3n$. claim $T(n)$ is $O(n^2)$

① pick $f(n) = n^2$

② pick 2 constants s.t.

for $n \geq n_0$, $5n^2 + 3n \leq c \cdot n^2$

Let $c = 1000$

and $n_0 = 1$

$c = 6$

$n_0 = 1$ ✗

Then the definition is true! ✓

claim $T(n)$ is $O(n)$

② pick c and n_0

$$5n^2 + 3n \le c \cdot n$$

$c = 1000$

$$5n^2 + 3n \le 1000 \cdot n$$

---

claim $T(n)$ is $O(n^3)$ ✓

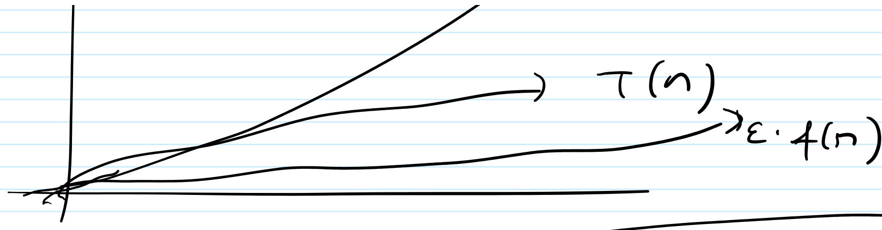
$\boxed{T(n) \text{ is } O(n^3)}$ ✓

"Searching a list takes $O(n)$ time"

Def We say $T(n)$ is $\Omega(f(n))$
if there exist $\varepsilon > 0$, $n_0 \ge 0$
where:
for all $n \ge n_0$, we have $T(n) \ge \varepsilon \cdot f(n)$

"searching takes $\Omega(n)$ time"

Def We say that $T(n)$ is $\Theta(f(n))$
if $T(n)$ is $O(f(n))$ and $\Omega(f(n))$


T(n)
ε·f(n)