

Runtime

```
foo(arr)
do
  swapped = false
  for i = 1 to len(arr)-1
    if arr[i-1] > arr[i]
      swap arr[i-1] and arr[i]
      swapped = true
  while swapped
```

Q: What is the runtime?

WC: $\Theta(n^2)$ - all swaps

BC: $\Theta(n)$ - no swaps

AC: $\Theta(n^2)$ - half swaps

Q: What is n ?

Array size

Graph size (edges vs vertices)

of bits

Q: Why do we care about runtime? Why not real execution time?

Q: What about space?

foo uses $\Theta(n)$ space and $\Theta(1)$ auxillary.

What counts as an atomic operation outside of bitland?

- Comparison

- Assignment

- Arithmetic operations (not exponentiation)

Q: why not?

Note that we ignore memory management overhead.

We're interested only in the complexity of operations on data structures. |

Big O notation

We only care about asymptotic behavior.

$$2x \quad 5x + 4 \quad x + \sqrt{x}$$

These are all basically the same function eventually in performance.

We formalize this notion into equivalence classes.

Let $f, g: \mathbb{N} \rightarrow \mathbb{N}$. We say $f \in O(g)$ if $\exists c \in \mathbb{R}^+ \forall^{\infty} n, |f(n)| \leq c|g(n)|$
 $\exists N \in \mathbb{N} : n \geq N \Rightarrow$

Ex) Prove $2n^2 + n - 1 \in O(n^2)$.

$$\begin{aligned} |2n^2 + n - 1| &\leq 2n^2 + n^2 - 1 && (n \geq 1) \\ &\leq 2n^2 + n^2 && (\text{everywhere}) \\ &= 3n^2 && (n \geq 1) \\ &= 3|n^2| && (n \geq 1) \end{aligned}$$

$$\boxed{c=3 \quad N=1}$$

Ex) Prove for $f, g: \mathbb{N} \rightarrow \mathbb{N}$ that if $f \in O(g)$, $f+g \in O(g)$. □

Since $f \in O(g)$, $\exists c_0 \in \mathbb{R}^+ \exists N_0 \in \mathbb{N} : |f(n)| \leq c_0|g(n)|$ when $n \geq N_0$.

Then when $n \geq N_0$,

$$\begin{aligned} |f(n) + g(n)| &\leq |f(n)| + |g(n)| \\ &\leq c_0|g(n)| + |g(n)| \\ &= (c_0 + 1)|g(n)|. \end{aligned}$$

$$\boxed{c=c_0+1 \quad N=N_0}$$

Obs) Smaller terms don't matter asymptotically. □

Ex) Prove $n^2 \notin O(n)$.

Let $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$ be given.

Then when $n \geq \max(c, N) + 1 > 0$

$$\begin{aligned} |n^2| &= n^2 \\ &> cn \\ &= c|n|, \end{aligned}$$

hence $n^2 \notin O(n)$.

□

Other Classifications

Big O

$f \in O(g) \approx f$ is at most g .

Little o

$f \in o(g)$ iff $\forall c \in \mathbb{R}^+ \forall^\infty n, |f(n)| \leq c|g(n)|$

$f \in o(g) \approx f$ is smaller than g .

Big Ω

$f \in \Omega(g)$ iff $\exists c \in \mathbb{R}^+ \forall^\infty n, |f(n)| \geq c|g(n)|$

$f \in \Omega(g) \approx f$ is at least g .

Little ω

$f \in \omega(g)$ iff $\forall c \in \mathbb{R}^+ \forall^\infty n, |f(n)| \geq c|g(n)|$

$f \in \omega(g) \approx f$ is bigger than g .

Theta

$f \in \Theta(g)$ iff $f \in O(g)$ and $f \in \Omega(g)$.

$f \in \Theta(g) \approx f$ is the same as g .

Complexity Relations

Come up with
some that seem
obvious

$$f \in o(g) \Rightarrow f \in O(g)$$

$$f \in \omega(g) \Rightarrow f \in \Omega(g)$$

$$f \in O(g) \Rightarrow g \in \Omega(f)$$

$$f \in o(g) \Rightarrow g \in \omega(f)$$

Limit Definitions

Consider the limit

$$\lim_{n \rightarrow \infty} \left| \frac{f(n)}{g(n)} \right| = m \quad \leftarrow \text{assuming the limit exists}$$

If the limit exists and $m = 0$, $f \in o(g)$.

If the limit exists and $m > 0$, $f \in \Theta(g)$.

Where does O fit into this?

If we flip f and g in the limit, we get the same results for ω, Ω .
(see complexity relations above).

How do we
get results
for ω, Ω ?