Week 9: Reminders

Problem set 4: Due Mon 22 (Next week)

Midterm 2: Next week (Mon 23/24)

Harder than usual
So start soon
Example - Runtime Analysis, not obvious

Variant of a famous function "3n+1" or Collatz function

Collatz(n):

\[ x = n \]
while \( x > 1 \):

if \( x \mod 2 = 0 \) then \( x = \frac{x}{2} \)
else \( x = 3x + 1 \)

Open: Does this algorithm terminate for every \( n \in \mathbb{N}^+ \)
Variation on Collatz that does always terminate

def f(n):
    x = n
    while x > 1:
        if x % 2 == 0 then x = \frac{x}{2}
        else x = 2x - 2

Our goal
Come up with a function h(n) s.t.
Runtime of f(n) = \Theta(h(n))
(Runtime is O(h(n)) and \Omega(h(n)))
1. Come up with $h_1(n)$ s.t.
   Runtime $f(n) \in O(h_1(n))$
   \[
   \exists n_0 \in \mathbb{N} \exists c_0 \in \mathbb{N} \quad (n \geq n_0 \Rightarrow \text{Runtime of } f(n) \leq c_0 \cdot h_1(n))
   \]

2. Come up with $h_2(n)$ s.t.
   Runtime $f(n) \in \Omega(h_2(n))$
   \[
   \exists n_0 \in \mathbb{N} \exists c_0 \quad (n \geq n_0 \Rightarrow \text{Runtime of } f(n) \geq c_0 \cdot h_2(n))
   \]

Ideally $h_1(n) \approx h_2(n)$ so can conclude $f(n) \in \Theta(h_1(n))$. But not always possible.
Let's do some examples.

\[ n = 10 \]
\[ x = 10 \]
\[ x = 5 \]
\[ x = 8 \]
\[ x = 4 \]
\[ x = 2 \]
\[ x = 1 \]

\[ x \text{ even } \Rightarrow \frac{x}{2} \]
\[ x \text{ odd } \Rightarrow 2x - 2 \]

\[ n = 13 \]
\[ \rightarrow x = 13 \]
\[ x = 24 \]
\[ x = 12 \]
\[ x = 6 \]
\[ x = 3 \]
\[ x = 4 \]
\[ x = 2 \]
\[ x = 1 \]

\( n \) is a power of 2

\[ x = 16 \]
\[ x = 8 \]
\[ x = 4 \]
\[ x = 2 \]
\[ x = 1 \]

Then runtime is \( \log_2 n \)
Look at 2 consecutive executions of the while loop.

\[ x = \text{odd} \]
\[ x \rightarrow 2x - 2 \]
\[ x \rightarrow \frac{2x - 2}{2} = x - 1 \]

\[ x = \text{even} \]
\[ x \rightarrow \frac{x}{2} \]
\[ \text{even} \rightarrow \frac{x}{4} \]
\[ \text{odd} \rightarrow 2 \left( \frac{x}{2} \right) - 2 = x - 2 \]
So after 2 executions of while loop, x goes down by at least 1.

So can conclude

Run-time \( (n) \leq 2n \)

\[ \text{s.t. } \text{Run-time } (n) \in O(n) \]
2. Now we want to show

\[ \text{Runtime}(n) \in \Omega(\log n) \]

\[ \in \Omega(\log_2 n) \]

\( \forall n \geq n_0, \forall n \geq 1, \text{Runtime}(n) \geq c \cdot \log_2 n \)

Pick \( c = 1 \)

Every time we go through the while loop, \( n \) decreases by at most \( \frac{3}{2} \)

So we get to \( x = 1 \) after \( \frac{n}{2^k} = 1 \)

\( k = \log_2 n \)
So far we know

\[ \text{Runtime}_f(n) \leq O(n) \]
\[ \text{Runtime}_\ell(n) \leq \Omega(\log_2 n) \]

Q: Can we do better?

O(n) is not tight!

Real answer is \( \Theta(\log_2 n) \)

Hint: Look at 3 consecutive executions of the loop
So far all programs that we analyzed had just one input for each site $n$.

Now: Look at algorithms where there are many different inputs of size $n$.

We have to decide how to measure runtime for all inputs of size $n$.

2 possibilities today: worst-case, best-case (also average-case that we will do later)
Worst-case Runtime of $f(x)$ over all $x$ of size $n$:

$$WC_{f}(n) = \max \left\{ \text{Runtime of } f(x) \mid x \text{ has size } n \right\}$$

Best-case Runtime:

$$BC_{f}(n) = \min \left\{ \text{Runtime of } f(x) \mid x \text{ has size } n \right\}$$
Example

```
Def even(numbers):
    for number in numbers:
        if number % 2 == 0:
            return True
    return False
```

Worst case runtime:

1. $W_{even}(n) \in O(n)$:

   For all $x$ (if $x$ has size $n \Rightarrow \text{Runtime of even on } x \leq n$)
Argument: Loop never executes more than $n$ times, on any input $x$ of size $n$.

$\forall n \in \mathbb{N}_0$
$\forall x \left( n \geq n_0 \text{ and } x \text{ has size } n \Rightarrow \text{Runtime of } \text{Gen}(x) \leq C_0 \cdot n \right)$

(2) $\text{WCC}_{\text{gen}}(n) \in \mathcal{O}(n)$:

$\forall n \in \mathbb{N}_0$
$\exists x \left( x \text{ has size } n \Rightarrow \text{Runtime of } \text{Gen}(x) \geq C_0 \cdot n \right)$

Take $x = (3,3,\ldots,3)$ a list of $n$ 3's (a list of $n$ odd numbers also works).
\[ WC_f(n) \in O(g_1(n)): \text{ show } \forall n \in \mathbb{N}_0 \text{ s.t. } n \geq n_0 \text{ and } \exists x \text{ of size } n \Rightarrow \text{“Runtime}_f(x) \leq g_1(n) \]

\[ WC_f(n) \in \Omega(g_2(n)): \text{ show } \forall n \in \mathbb{N}_0 \text{ s.t. } n \geq n_0 \text{ and } \exists x \text{ of size } n \text{ and Runtime}_f(x) \geq g_2(n) \]

\[ BC_f(n) \in O(g_3(n)): \text{ show } \forall n \in \mathbb{N}_0 \text{ s.t. } n \geq n_0 \text{ and } \exists x \text{ of size } n \text{ and Runtime}(x) \leq g_3(n) \]

\[ BC_f(n) \in \Omega(g_4(n)): \text{ show } \forall n \in \mathbb{N}_0 \text{ s.t. } n \geq n_0 \text{ and } \forall x \text{ of size } n \Rightarrow \text{Runtime}(x) \geq g_4(n) \]
Best case complexity of Even:

1. $B_{\text{even}}(n) \in O(\frac{1}{n})$

   $\forall n \exists x \ (\text{length } n + \text{runtime} \leq \frac{1}{n})$

   Let $x = (\text{even number, } \star, \star, \star, \ldots \star)$

   ex. $x = (\frac{1}{4}, 2, 2, 2, 2, 2, \ldots)$

2. $B_{\text{even}}(n) \in \Omega(\frac{1}{n})$

   $\forall n \forall x \ (x \text{ has length } n \Rightarrow \text{runtime is } \geq \frac{1}{n})$

   $\therefore B_{\text{even}}(n) = \Theta(1)$
One more example.

A palindrome is a string (a sequence of letters) that reads the same forwards as backwards.

aaaabaaa
MOM
DAD
aabcbaa
aabcbaa

Not a palindrome

palindromes
```python
def Pal(s):
    n = len(s)
    for k in range(n, 1, -1):
        Is_pal = True
        for j in range(k-1, -1, -1):
            if s[j] != s[k-j-1]:
                Is_pal = False
                break
        if Is_pal:
            return j
```

A more efficient implementation would stop at `j = \frac{k}{2}`.
What is worst case complexity?

Upper bound:

On every input $s$ of length $n$ the outer loop executes at most $n$ times the inner loop executes at most:

$$n + n-1 + \ldots + 1$$

$$= \frac{n(n+1)}{2} = O(n^2)$$

$$\forall n \forall s \ (s \text{ has length } n) \Rightarrow \text{Runtime}_{pol}(s) \leq 2n^2$$
What about $W_{\text{pal}}(n) \in \Omega(n^2)$?

Need to find an input $S$ such that $S$ has length $n$ and runtime on $S \geq c \cdot n^2$

Fix $n$. Let $S$ start with one $a$, followed by $n-1$ $b$'s

$a_{\underbrace{b bb b b b b b b b}} \iff n=10$

The runtime on this string is $\sim \frac{n(n+1)}{2}$

$= \Omega(n^2)$
alternatively we could have defined $s$ as

all $a$'s, with a one $b$ in position

$= \left\lceil \frac{n}{2} \right\rceil + 1$

$n = 10$

$\underline{a a a a a b a a a a}$

$n = 11$

$\underline{a a a a a a a a a a b a a a a}$

runtime on this sequence of $s$'s is $\geq \frac{n}{2} \cdot \frac{n}{2} = \mathcal{O}(n^2)$
So \( WC_{\text{pal}}(n) = \Theta(n^2) \)!

How about Best case runtime? The best case runtime is \( \Theta(n) \)

1. \( BC_{\text{pal}}(n) \in O(n) \) [For every \( n \), I can find a string \( s \) of length \( n \) whose runtime is \( \leq n \)]

\( \forall n \exists s \) (\( s \) has length \( n \) and \( \text{Runtime}_{\text{pal}}(s) \leq n \))

Let \( s \) be all a's

i.e. \( n=8 \) \( s = \text{aaaaaaaaa} \)

2. \( BC_{\text{pal}}(n) \in \Omega(n) \) [For every \( n \), I can't find a string \( s \) that length that has runtime \( \leq n \)]

\( \forall n \forall s \) (\( s \) has length \( n \) ⇒ runtime \( (s) \geq n \))
Alg stupid (s)

Print s[0]

exit

s string

n = length of s

* Usually runtime depends on n, except in degenerate cases

An example where

worst case runtime = best case runtime = Θ(1), since runtime doesn't depend on n