- Reminder: beware of online sources (like Chegg) and always cite your sources!
- PS3 general extension — see announcements!

So far...

- come up with "exact" count of steps/runtime
- use $O/\Omega/\Theta$ to simplify

Starting now: no "exact" count — come up with suitable Θ-expression directly!

Example 1:

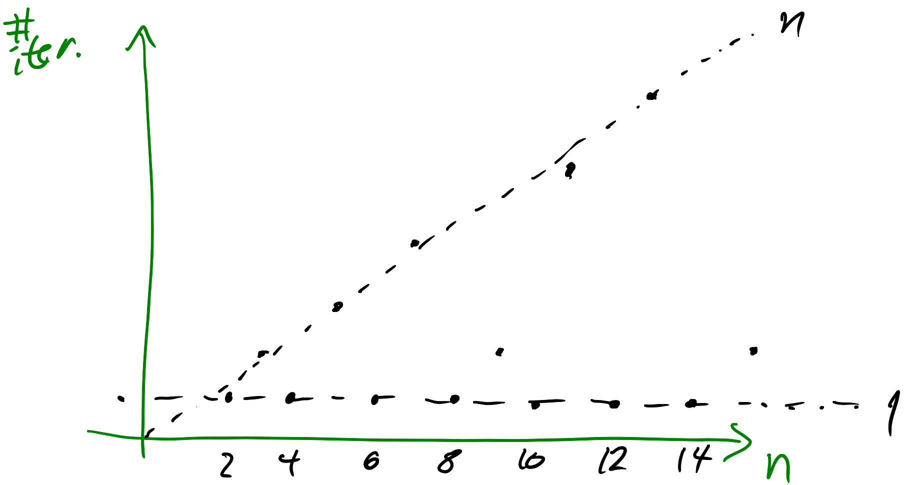
Runtime?

```
0. def is_prime(n: int) -> bool:
1.     for d in range(2, n):
2.         if n % d == 0:
3.             return False
4.     return True
```

- # iterations of loop? — ?
- steps for loop body? — lines 2-3: 1 step
- steps outside loop? — line 4: 1 step

for every input n,

- # iterations ≥ 1 (when n is even)
- # iterations $\leq n-2$ (# of values in range(2,n); when n is prime)



Conclusion:

- runtime is $\Omega(1)$
- runtime is $O(n)$
- no simple Θ

<u>NOTATION</u>:

"$RT_A(\underline{x})$" is the running time (# steps) of algorithm A on <u>input x</u>

<u>NOTE</u>: RT is a function of the <u>input</u>, not just the input <u>size</u>.

Conclusion for is_prime:

$$\left.\begin{array}{l} RT_{ip}(n) \in \Omega(1) \\ \wedge \ RT_{ip}(n) \in O(n) \end{array}\right\} \underline{\text{NO}}\ \Theta \text{ for } RT_{ip}$$

<u>Example 2</u>:

```
0. def print_primes(n: int) -> None:
1.     for k in range(2, n+1):
2.         if is_prime(k):
3.             print(k)
```

$RT_{pp}(n)$?

Observation: don't expect to find a simple Θ-expression. So split up analysis into upper bound & lower bound.

Upper bound (O) — overestimate

- Loop body (lines 2–3) takes time $O(k) \le n$
- # iterations $\le n$
- total $\le n^2$ $\Rightarrow$ $O(n^2)$

Lower bound (Ω) — underestimate

- loop body takes time $\Omega(1) \geq 1$
- # iterations $\geq n-1$
- total $\geq n-1 \implies \Omega(n)$

More careful calculation...

– upper bound:

$$RT_{pp}(n) = \sum_{k=2}^{n} RT_{ip}(k)$$

(note: really, it's more $\sum_{k=2}^{n} (RT_{ip}(k) + 1)$

$$= \left(\sum_{k=2}^{n} RT_{ip}(k)\right) + (n-1))$$

$$\leq \sum_{k=2}^{n} k \qquad (RT_{ip}(k) \in O(k))$$

$$\cong \frac{n^2}{2} \implies O(n^2)$$

- lower bound:

$$RT_{pp}(n) = \sum_{k=2}^{n} RT_{ip}(k)$$

$$= \left(\sum_{\substack{2 \leq k \leq n \\ k \text{ is prime}}} RT_{ip}(k) \right) + \left(\sum_{\substack{2 \leq k \leq n \\ k \text{ is not prime}}} RT_{ip}(k) \right)$$

$RT_{ip}(k) \in \Omega(k)$ when k is prime

$$\geq \boxed{\sum_{\substack{2 \leq k \leq n \\ k \text{ is prime}}} k}$$

NOT OBVIOUS — you would need to look this up, but it turns out that $\sum$ prime numbers $\leq n \;\widehat{=}\; \frac{n^2}{\log n}$

$$\geq \frac{n^2}{\log n} \implies \boxed{RT_{pp}(n) \in \Omega\left(\frac{n^2}{\log n}\right)}$$

<u>still not equal</u> to $O(n^2)$, but closer...