

## 1 An Example

Examine the code:

```
1 def is_prime(n: int) -> bool:    #precondition: n >= 2
2     for d in range(2, n):        #What number of iterations here?
3         if n % d == 0:           #1 Step
4             return False        #1 Step
5     return True                  #1 Step (if we get here)
```

Examining the number of iterations, we notice that it is very dependent on *if* the number is in fact prime with even numbers returning after just 1 iterations and most returning under 5.

Therefore,  $RT_{ip}(n) = \#iterations + maybe1$   
so,  $RT_{ip}(n) \in \mathcal{O}(n)$  and  $RT_{ip}(n) \in \Omega(1)$

```
1 def print_primes(n: int) -> None    #precondition: n >= 2
2     for k in range(2, n + 1):        #How many iterations: (n - 1)
3         if is_prime(k):              #Number of steps <= ck <= cn
4             print(k)
```

Therefore,  $(n - 1)c_2 \in \Omega(n) \leq RT_{pp} \leq (n - 1)cn \in \mathcal{O}(n^2)$