1 An Example

Examine the code:

```
def twisty(n: int) -> None:      #Precondition: n >= 1
    while n > 1:
        if n % 2 == 0:
            n = n//2
        else:
            n = 2 * n - 2
```

Examining the function above with the following inputs to `n`:

`n`	Values of `n`
1	1
2	2, 1
3	3, 4, 2, 1
4	4, 2, 1
5	5, 8, 4, 2, 1
165	165, 328

We see:

- if `n` is odd, then $2 * n - 2$ is even, and therefore after 2 iterations, it's $n - 1$
- If `n` is even, *and* $n > 1$ then after 1 iteration it's $\frac{n}{2} \leq n - 1$

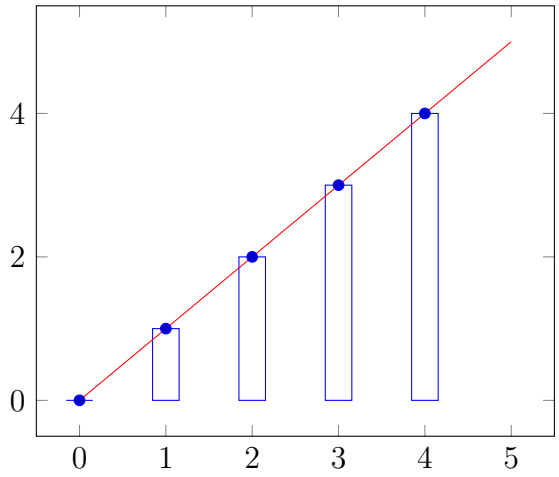
Therefore, we note that every 1 or 2 iterations, n goes down by at least 1 and concequently the program must terminate after at most $2n$ iterations. In this way $RT_{twisty} \in \mathcal{O}(n)$

1.1 Non integer inputs

```
def is_pal(s: str) -> bool:
    for i in range(len(s)):
        if s[i] != s[len(s) - 1 - i]:
            return False
    return True
```

We note that the runtime is a function of the input - which is a string - so we must determine a method for measuring the input in natural numbers. In this case, we use the size/length.

Potential worst case runtime of `is_pal` given different sized inputs



For $n \in \mathbb{N}$, let $\mathcal{I}_n = \{s \in str : |s| = n\}$ Let $WC(n) = max\{RT(s) : s \in \mathcal{I}_n\}$

Let $E \subseteq \mathbb{R}$ and E has a maximum and $l \leq maxE \leq u$

it's worth noting that $maxE \in E$ and $\forall e \in E, e \leq maxE$