CSC165 fall 2019

worst/best/average

Danny Heap
csc165-2019-09@cs.toronto.edu
BA4270 (behind elevators)
Web page:
http://www.teach.cs.toronto.edu/~heap/165/F19/

Using Course notes: algorithm analysis

frequently asked...

```
def all_pairs(lst: list) -> None:
    i = 0
    while i < len(lst):
        j = 0
        while j < i:
            print(i + j)
            j = j + 1
        i = i + 1
```

$$RT_{all\_pairs}(n) = \sum_{i=0}^{n-1}(i+1) = \sum_{i'=1}^{n} i' = \frac{n(n+1)}{2} + n \in \Theta(n^2).$$

compare...

```
def is_prime(n):
    if n < 2:
        return False
    else:
        for d in range(2,n):
            if n % d == 0:
                return False
        return True
```

```
def has_even(number_list):
    for number in number_list:
        if number % 2 == 0:
            return True
    return False
```

definitions

- $\mathcal{I}_{f,n} = \{i \mid i \text{ is an input to } f \wedge |i| = n\}$
- $RT_{f(x)} =$ number of basic "steps" in executing $f(x)$
- $WC_f(n) = \max\{RT_{f(x)} \mid x \in \mathcal{I}_{f,n}\}$

upper bounds, lower bounds...

- $U(n)$ is an upper bound means
  $\forall n \in \mathbb{N}, \forall x \in \mathcal{I}_{f,n}, RT_{f(x)} \leq U(n)$

- $L(n)$ is a lower bound means
  $\forall n \in \mathbb{N}, \exists x \in \mathcal{I}_{f,n}, RT_{f(x)} \geq L(n)$

why the asymmetry of U and L?

$WC_{\text{has\_even}} \in O(n)$

$WC_{\text{has\_even}} \in \Omega(n)$

palindromes

examples: “racecar rotor pap…” every string **starts** with a palindrome, so find the longest palindrome prefix…

```
def palindrome_prefix(s):
    n = len(s)
    for prefix_length in range(n, 0, -1): # count down from n
        is_palindrome = True
        for i in range(prefix_length):
            if s[i] != s[prefix_length - 1 - i]:
                is_palindrome = False
                break
        if is_palindrome:
            return prefix_length
```

average...

$$\mathcal{I}_{f,n} = \{i \mid i \text{ is an input to } f \wedge |i| = n\}$$
$$\mathcal{T}_{f,n} = \{t \mid \exists x \in \mathcal{I}_{f,n}, t = RT_f(x)\}$$

```
def has_even(number_list):
    for number in number_list:
        if number % 2 == 0:
            return True
    return False
```