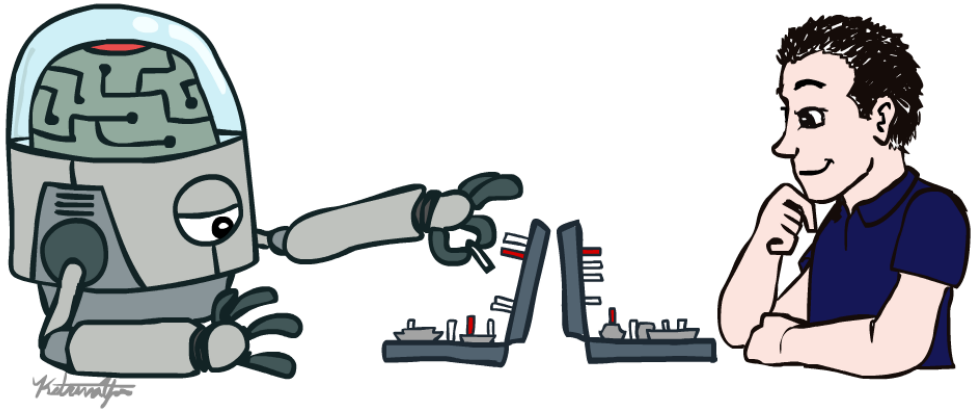
Big O



Ashis Kumar Chanda
chanda@rowan.edu

RowanUniversity

Introduction

- Big O notation is used to classify algorithms according to how their run time grow as the input size grows.
- In other term, **how long an algorithm takes to run.**
- The letter O is used because the growth rate of a function is also referred to as the **order of the function**.
- Big O notation usually only provides an **upper bound** on the growth rate of the function.

Example

- What is the big O of the following operation:
- X = 10 + 5 x n
- Suppose, n is any real number.
- Look, the value of n doesn't affect processing of code/algorithm.
- So, it is O(1).

Example

- What is the big O of the following operation:

- for (i=0; i<n; i++){

  X += 10 + 5 x i

  }

- Suppose, n is any real number and n >1.

- Look, the operation is running for n times.
- So, it is O(n).

Example

- What is the big O of the following operation:

- for (i=0; i<n; i++){

```
 for (j=0; j<i; j++){
      X += 10 + 5 x j
      }
 }
```

- Suppose, n is any real number and n >1.

n	operations
2	1
3	1+2
4	1+2+3
…	…
n	1+2+3+… +(n-1)

- Look, the operation is running for n*(n-1)/2 times.
- So, it is O(n^2).

Example

- What is the big O of the following operation:

- for (i=1; i<n; i= i*2){
  print (i)
  }

- Suppose, n is any real number and n = 16.

- So, it is $O(\log_2(n))$.

step	print
1	1
2	2
3	4
4	8
…	…
h	$\text{Log}_2(n)$

Example

- What is the big O of the following operation:

- for (i=n; i>=1; i= i/2){
  print (i)
  }
- Suppose, n is any real number and n > 1.

- So, it is $O(\log_2(n))$.

Suppose, n is 16

step	print
1	16
2	8
3	4
4	2
5	1

Notes

- In typical usage the O notation is asymptotical, that is, it refers to very large input values.
- As a result, the following simplification rules can be applied:
  - If f(x) is a sum of several terms, if there is one with largest growth rate, it can be kept, and all others omitted.
  - If f(x) is a product of several factors, any constants (terms in the product that do not depend on x) can be omitted.

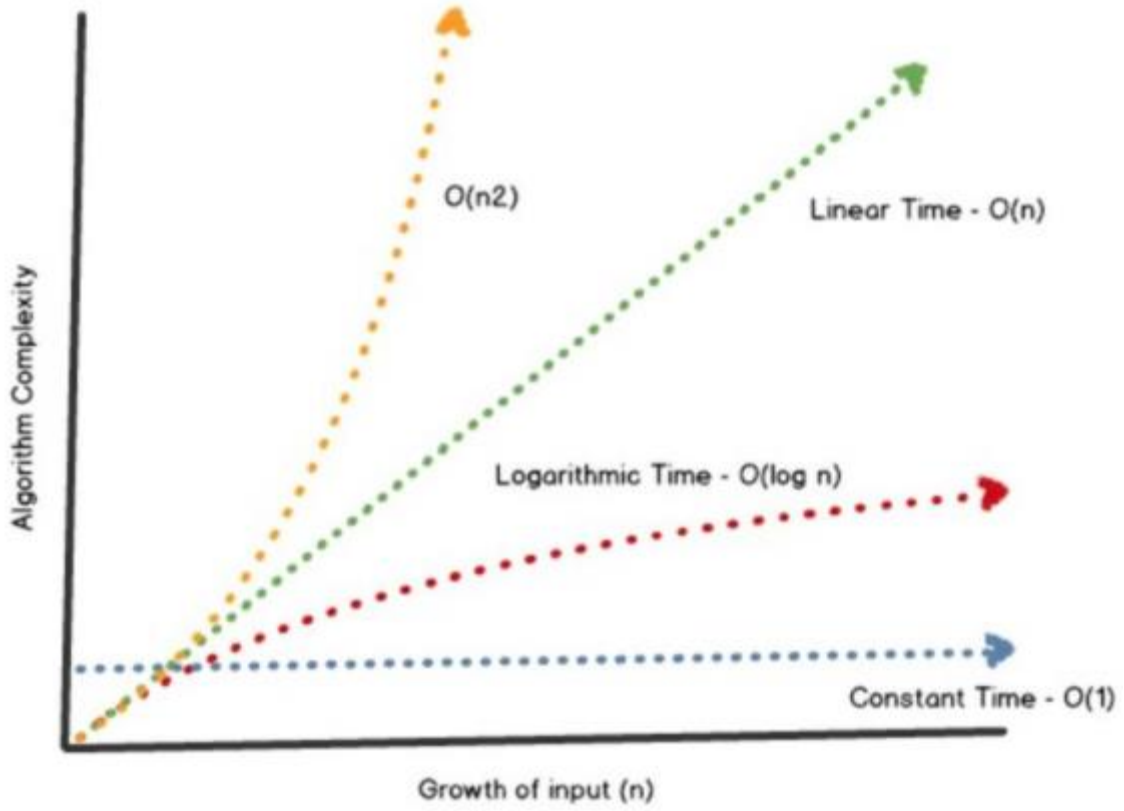
Example

- What is the big O of the following operation:

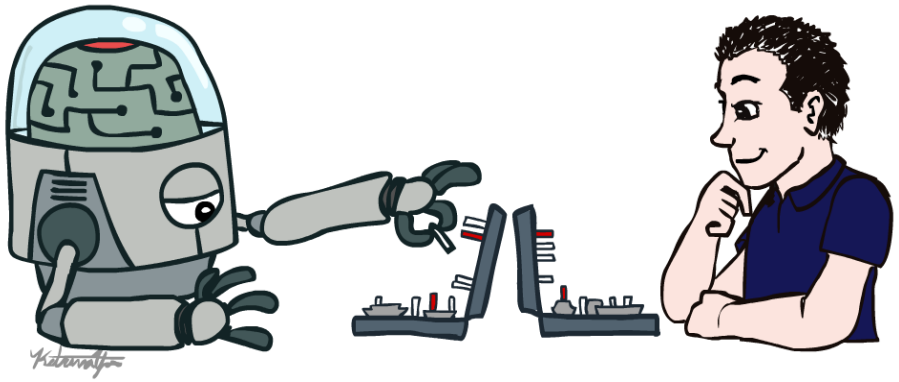
- F(x) = $x^4 - 2x + 5$
- So, it is O(n^4).

- F(x) = $(6x^4 - 2)/2$
- So, it is O(n^4).

Different time complexity



Graphs of functions commonly used in the analysis of algorithms



Thanks!