Data structures and algorithms

Getting Interviews

Big O Notation – How long it takes an algorithm to run

As input grows, how much does the algorithm slow down

What is good code?

-Readable and scalable

Read- is it clean and easy to read

Scalable- Big O is used to measure this

Think of baking a cake. Ingredients -> kitchen -> cake

Code(instructions) -> computer -> result. Coders give the instructions

Runtime – how long something takes to run

O(n) 🡪 Linear time

proportional

Example O(1), O(10), O(10000)

N can be anything i.e. x, fish etc

Big O focuses on how fast a runtime grows

O(1) -> constant time

Excellent (very scalable)

1 operation no matter how big the input is

10000 items in an array vs 1 item in array will both have 1 operation

Thing that happen once O(1) like assignment

Things that happen repeatedly O(n) like for loops and if statements

Simplifying Big O – 4 Rules

1. Worst case

Assume worst ccase

In a if loop, assume all the loops will occur