<image/nation>

Tech Training Series

Exclusive for HKU BUSINESS SCHOOL 港大經管學院

Data Structure & Algorithm

2024/2025

How data are stored and how data are processed efficiently

by

Sunny NG

<image/nation>

To download the slides

`bit.ly/in-download`

In this workshop (3 hours)

- Introduction to algorithm
- Introduction to data structure
- Algorithm Examples
- Data structure and algorithm visual simulation
- Common data operations
- Computation Complexity
- Time vs. Space (Speed vs. Storage)

- **Hands-on Practicing**
  - Variable, array, objects and JSON
  - Looping
  - Array iterating
  - Built-in methods for Array and object
  - Searching algorithm
  - Sorting algorithm
  - Map/Reduce

Sunny Ng 吳新陸

Master of Fine Art, **CityU (HK)**
Master of Science, **HKU**
Bachelor of Science, **UH (UK)**

- Founder / Master Trainer Image Nation
- Part-time Lecturer HKU, HKUSPACE, City University of Hong Kong
- Developer AI, Web, Mobile, WeChat & IoT
- Content Creator Video producing / Live streaming
- Certified Azure AI Engineer
- AWS Solution Architect - Associate
- Alibaba Cloud Certified Professional
- AWS Academy Educator
- Email: sunny.ng@imagenation.com.hk

in linkedin.com/in/ngsunny/

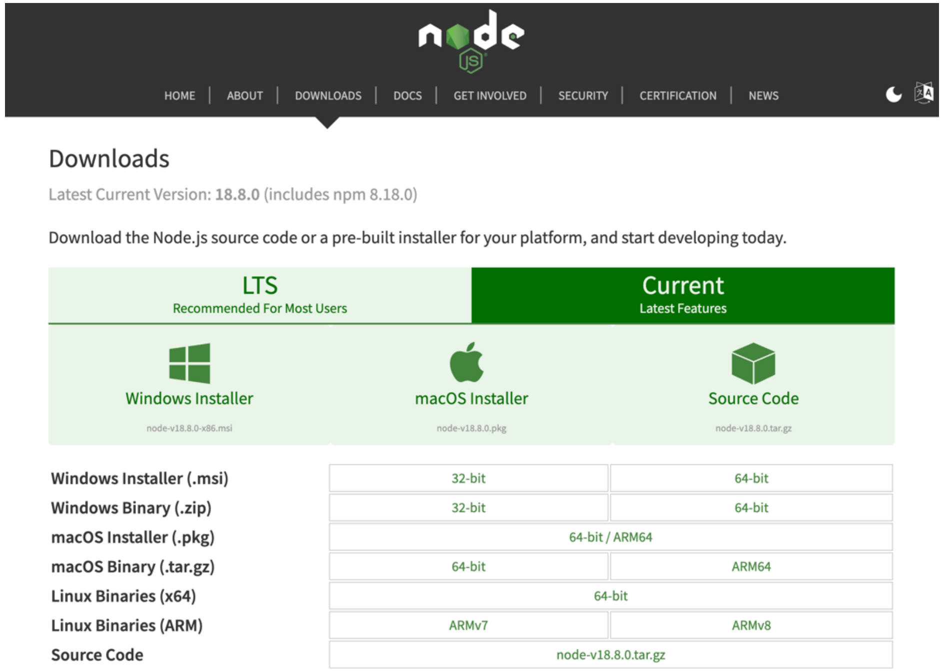
github.com/ngsanluk

Required Software Installation

1. Node JS
2. Visual Studio Code
3. Google Chrome Browser

Node.js Download

- Node.js is a popular development tool and runtime for JavaScript/Web Apps
- Download link
  - https://nodejs.org/en/download/



Downloads

Latest Current Version: **18.8.0** (includes npm 8.18.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

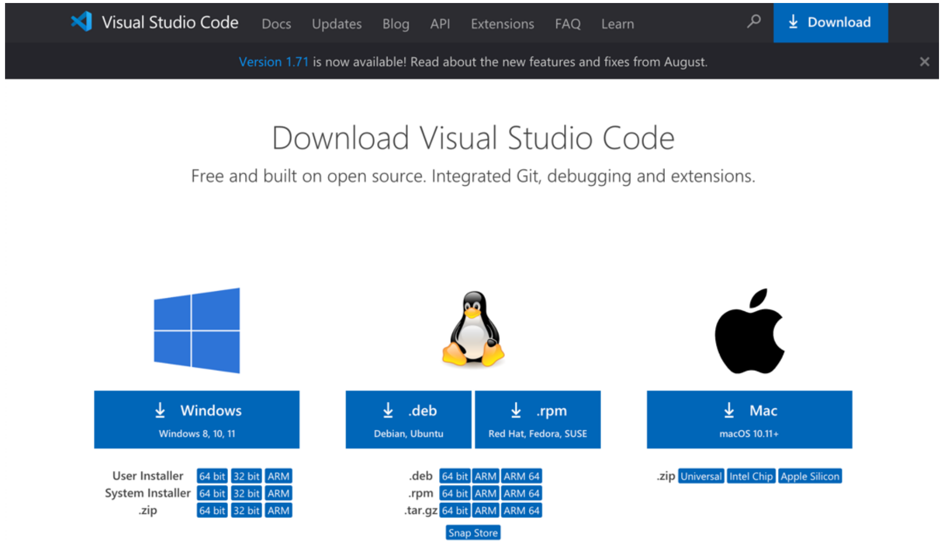
LTS Recommended For Most Users	Current Latest Features

Windows Installer node-v18.8.0-x86.msi	macOS Installer node-v18.8.0.pkg	Source Code node-v18.8.0.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v18.8.0.tar.gz	

Visual Studio Code

- Visual Studio Code is one of the most popular modern code editors
- We will use VS Code for JavaScript coding/editing
- Download link
  - https://code.visualstudio.com/download



Version 1.71 is now available! Read about the new features and fixes from August.

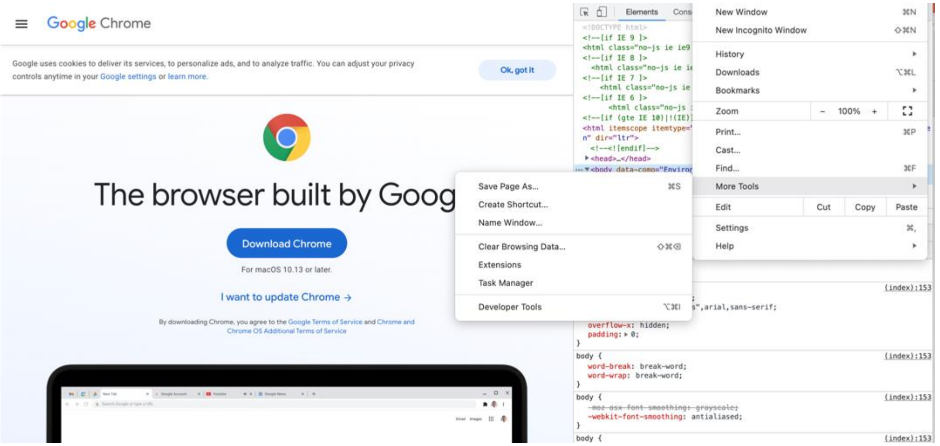
Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Google Chrome Browser

- We will need to use the Chrome Developer Tools
- Google “Chrome installer” to download and install

  or
- Visit download link
  - https://www.google.com/intl/en_hk/chrome/



Google uses cookies to deliver its services, to personalize ads, and to analyze traffic. You can adjust your privacy controls anytime in your Google settings or learn more.

The browser built by Goog

For macOS 10.13 or later.

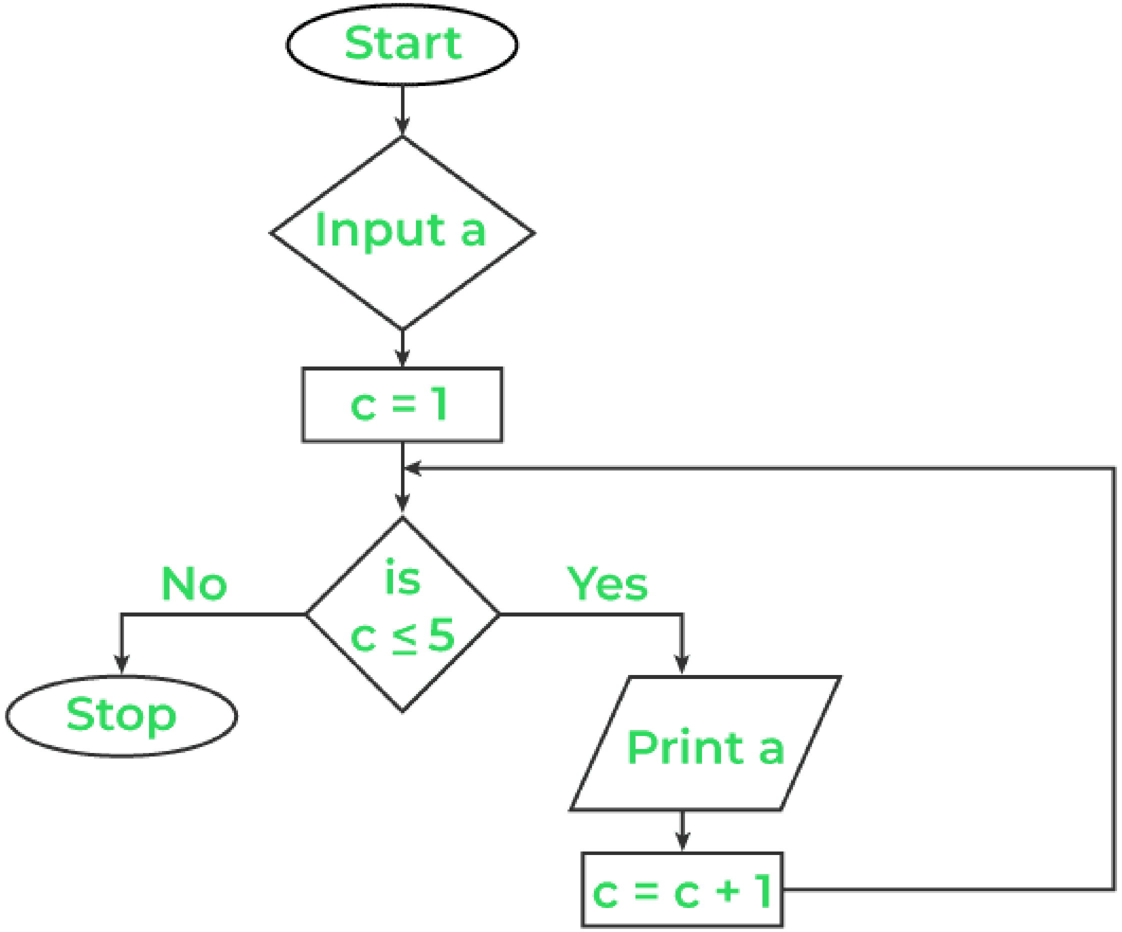
By downloading Chrome, you agree to the Google Terms of Service and Chrome and Chrome OS Additional Terms of Service

```
<!DOCTYPE html>
<!--[if IE 9 ]>
<html class="no-js ie ie9
<!--[if IE 8 ]>
  <html class="no-js ie ie
<!--[if IE 7 ]>
    <html class="no-js ie
<!--[if IE 6 ]>
      <html class="no-js
<!--[if (gte IE 10)|!(IE)]
<html itemscope itemtype="
n" dir="ltr">
  <!--<![endif]-->
 ▸<head>…</head>
…▾<body data-comp="Enviror
```

```
s",arial,sans-serif;
  overflow-x: hidden;
  padding: ▸ 0;
}
body {
  word-break: break-word;
  word-wrap: break-word;
}
body {
  ~~-moz-osx-font-smoothing: grayscale;~~
  -webkit-font-smoothing: antialiased;
}
body {
```

What is Algorithm?

Algorithm is the step-by-step procedure that defines a set of instructions to be executed in a certain order to get the desired output.



Searching has been a primary computation problem

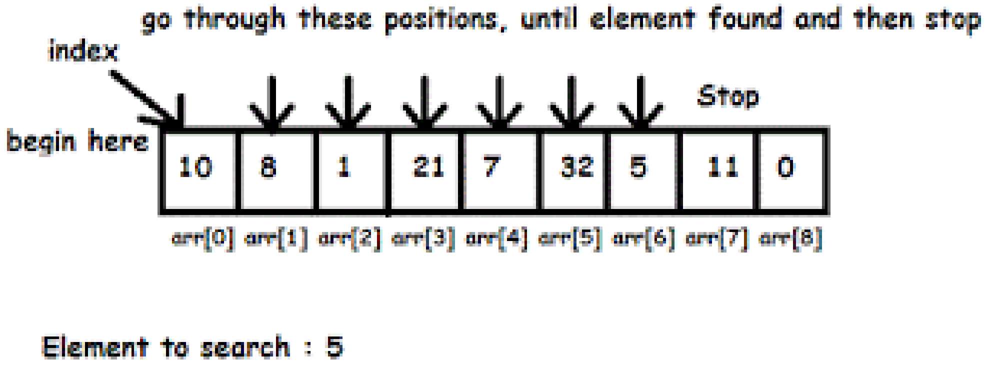
- Algorithms are there to solve computation problems while computation problems usually roots to daily life problems
- One of classical computation problems is **searching**

Search is also real-life problem: large collection of books

Searching is easy? **Yes** and **No**.

- Search is easy.
- But search efficiently is NOT easy at all.
- Consider the data volume that a nowadays business application is dealing with
  - No. of social media posts
  - No. of YouTube video uploads
  - No. of transactions on leading e-commerce platform

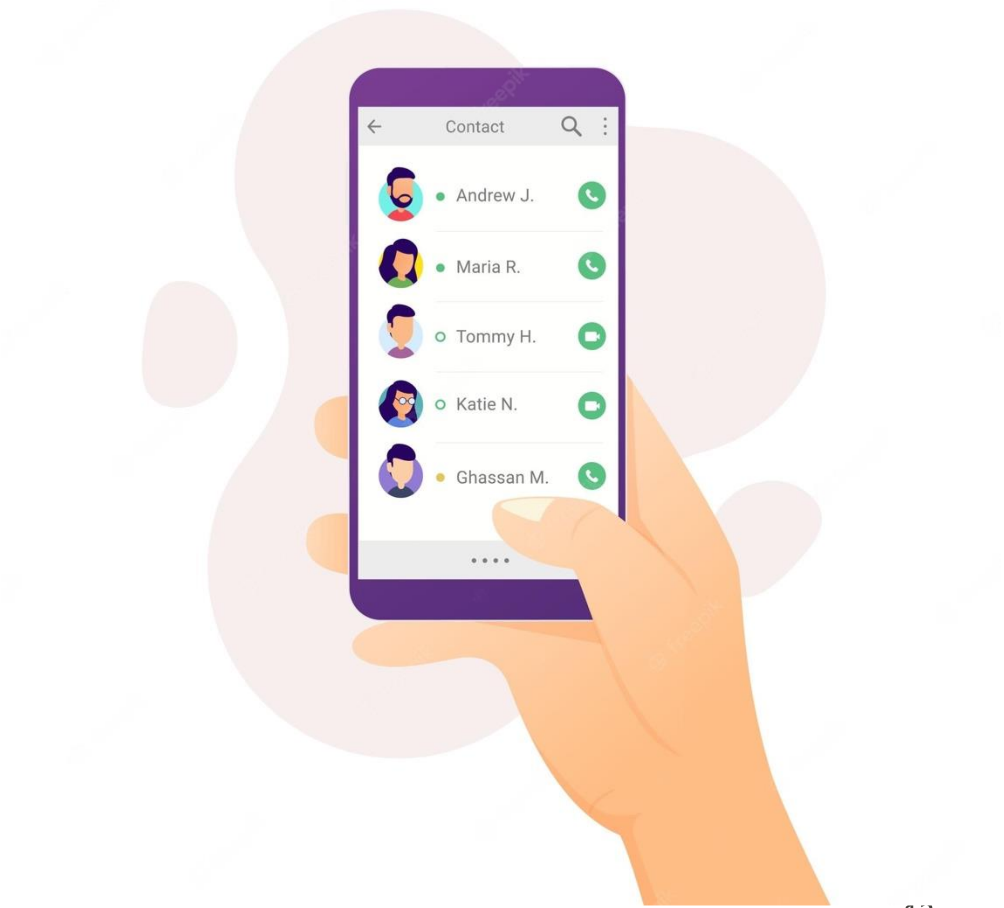
Linear search does give the answer, but way too slow



<image/nation>

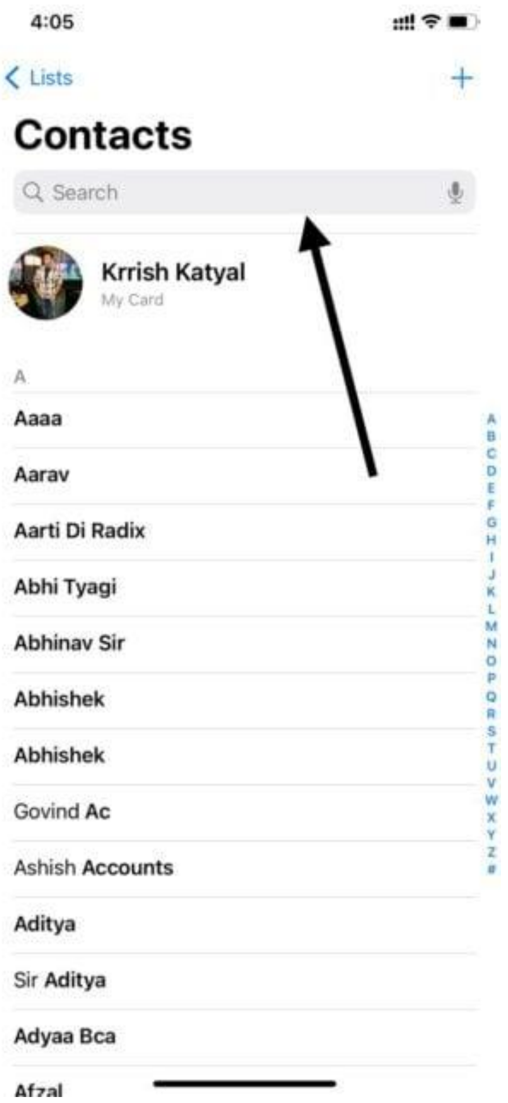
Phonebook browsing

- If there are only dozens of names in your phone book, **browsing** (linear search) the name list one-by-one might be okay.



Phonebook searching

- When phone book entries grow (imagine this is the clients address of an international big corp.), **searching** (by keyword) is the only answer
- Searching is a smart algorithm that requires data in certain order.



Contacts

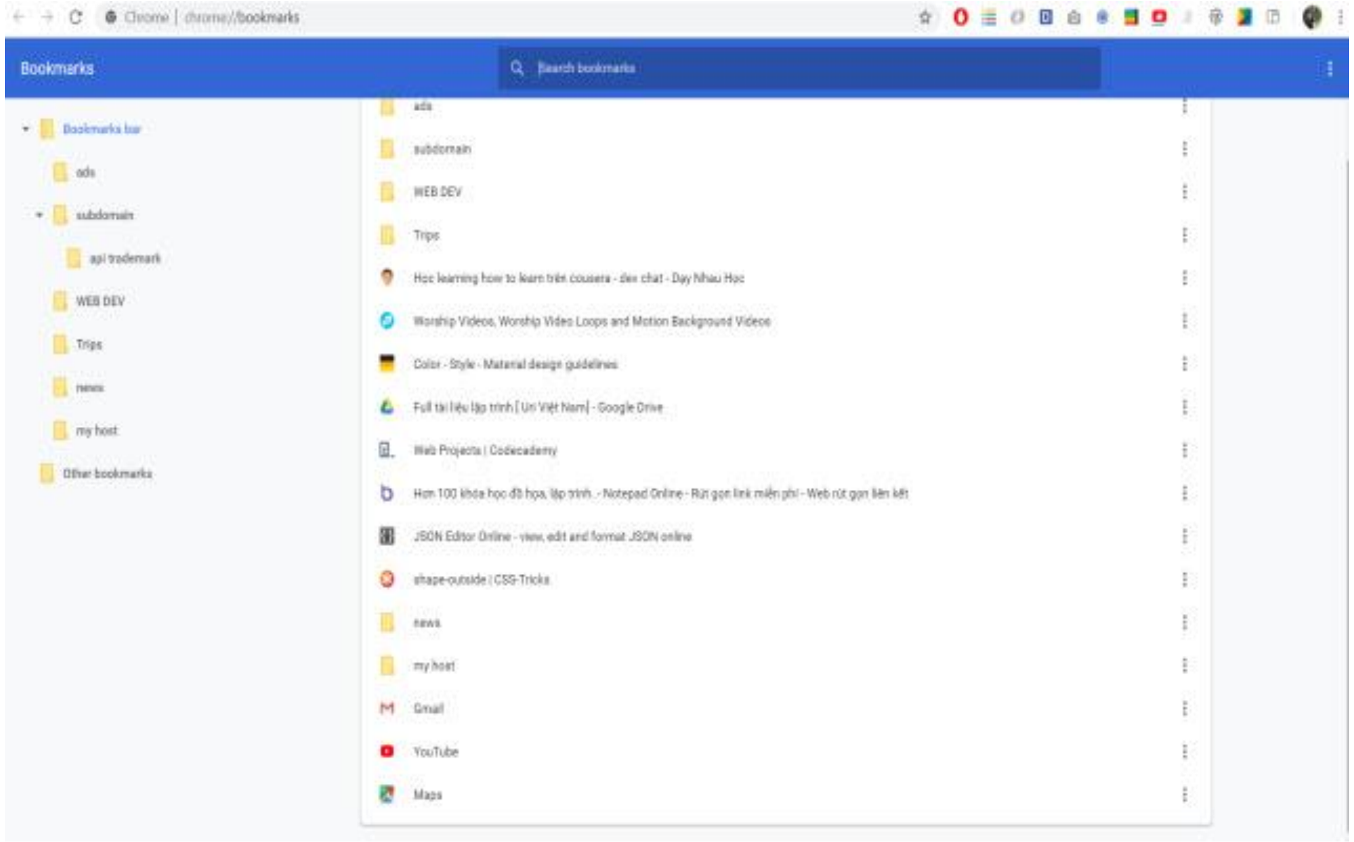
Krrish Katyal
My Card

A

Aaaa
Aarav
Aarti Di Radix
Abhi Tyagi
Abhinav Sir
Abhishek
Abhishek
Govind Ac
Ashish Accounts
Aditya
Sir Aditya
Adyaa Bca
Afzal

In the early days of the Web

- Each web user would build their own bookmarks to include websites of interest



Chrome | chrome://bookmarks

Bookmarks

Search bookmarks

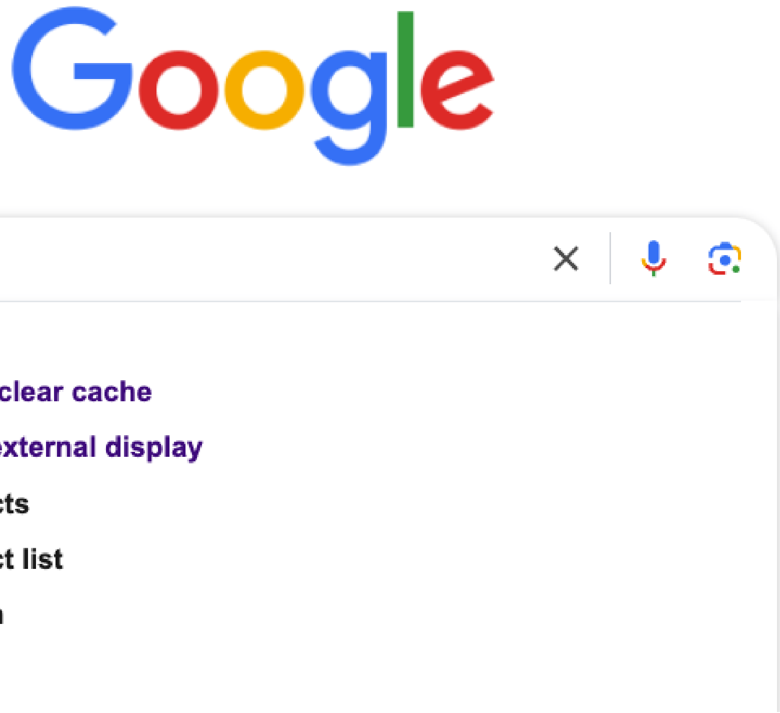
Bookmarks bar
ads
subdomain
api trademark
WEB DEV
Trips
news
my host
Other bookmarks

ads
subdomain
WEB DEV
Trips
Hoc learning how to learn trên cousera - dev chat - Dạy Nhau Học
Worship Videos, Worship Video Loops and Motion Background Videos
Color - Style - Material design guidelines
Full tài liệu lập trình [Un Việt Nam] - Google Drive
Web Projects | Codecademy
Hơn 100 khóa học đồ họa, lập trình.. - Notepad Online - Rút gọn link miễn phí - Web rút gọn liên kết
JSON Editor Online - view, edit and format JSON online
shape-outside | CSS-Tricks
news
my host
Gmail
YouTube
Maps

Now we search the web

- We search on Google and YouTube too many times each day
- Take a guess if I search “iPhone”, how many webpages on the web will match this keyword?

Let’s give it a try



iphone

iphone case

iphone safari clear cache

iphone 16 to external display

iphone contacts

iphone contact list

iphone search

iphone 15

Google Changes Life

- Do you realize how ridiculously **huge** is Google’s database?
- Do you also realize how **fast** Google are performing searching?

What is Google?

- Google search engine uses a complex algorithm called **PageRank** to determine the relevance and importance of web pages in its search results.
- PageRank analyses the number and quality of links pointing to a web page to assess its authority and popularity.
- **Bigtable**: Google's proprietary database system used for storing and managing large amounts of structured data.

Data size for modern computing

No. of 0s	Number Value	Computing Unit	Name in International System
3	1,000	Kilo	Thousand
6	1,000,000	Mega	Million
9	1,000,000,000	Giga	Billion
12	1,000,000,000,000	Tera	Trillion
15	1,000,000,000,000,000	Peta	Quadrillion
18	1,000,000,000,000,000,000	Exa	Quintillion
21	1,000,000,000,000,000,000,000	Zetta	Sextillion

- My computer drive is **1 Terabyte** (**1TB**).
- Big data is data storage that starts from **Petabyte** (**1PB**) and above.

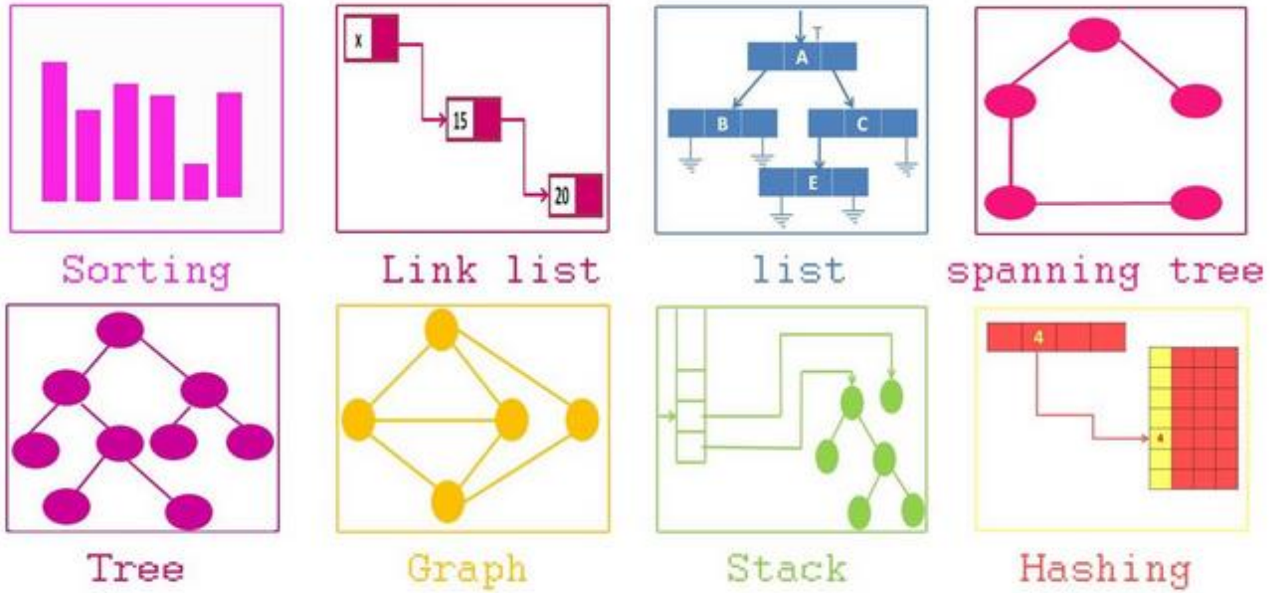
What is Data Structure?

- Data Structures are the **special programmatic way** of storing data so that data can be used efficiently.
- Data Structure and Algorithm form an optimized combination to solve computation problem **efficiently**.
  - Widely referred as DSA (Data Structure and Algorithm)
- Google gives their search result in such rapid speed because of their highly optimized data structure and algorithm and computation infrastructure.

<image/nation>

Common Data Structures

- Array
- Linked List
- Stack
- Queue
- Tree
- Hash-table
- Graph



Computer algorithms are written to be understood and executed by CPU.

Therefore, they could be difficult to understand by people without programming training

Swapping two number is straight forward to human brain

Say -

We have two numbers

- `x = 100`
- `y = 200`

A human brain would

Just get to the answer

- `x = y`
- `y = x`

Let’s try to code swapping in JavaScript

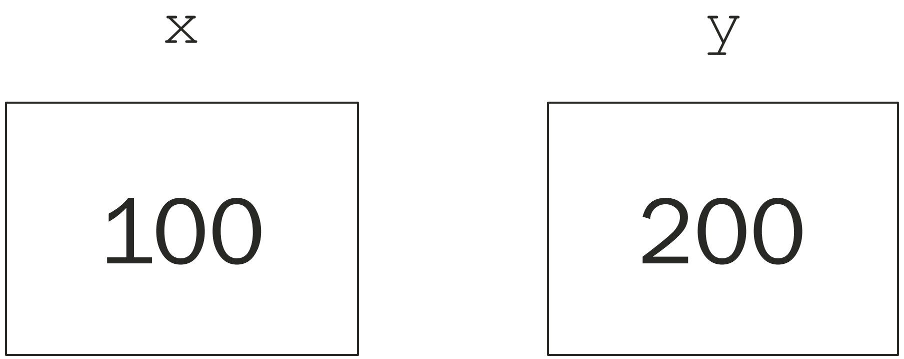
Unfortunately, the codes below DON’T work as we expect

```
x = 100;
y = 200;
console.log("Before Swapping");
console.log(`x = ${x}`);
console.log(`y = ${y}`);

// swapping x and y
x = y;
y = x;

console.log("\nAfter Swapping");
console.log(`x = ${x}`);
console.log(`y = ${y}`);
```

Swapping Algorithm

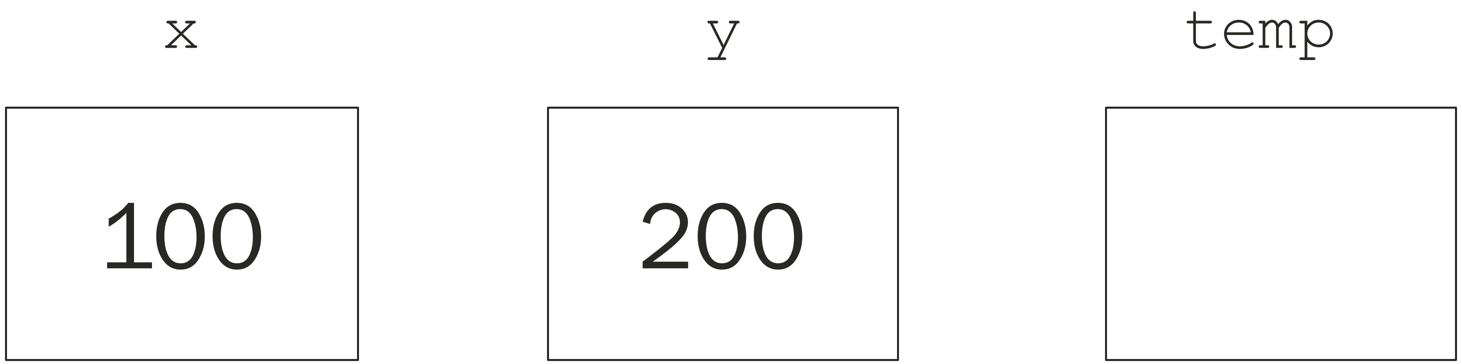
Imagine computer memory as lockers

`x` and `y` are the locker ID (memory address)



We need a temporary locker

`temp` is the temporary locker



x

100

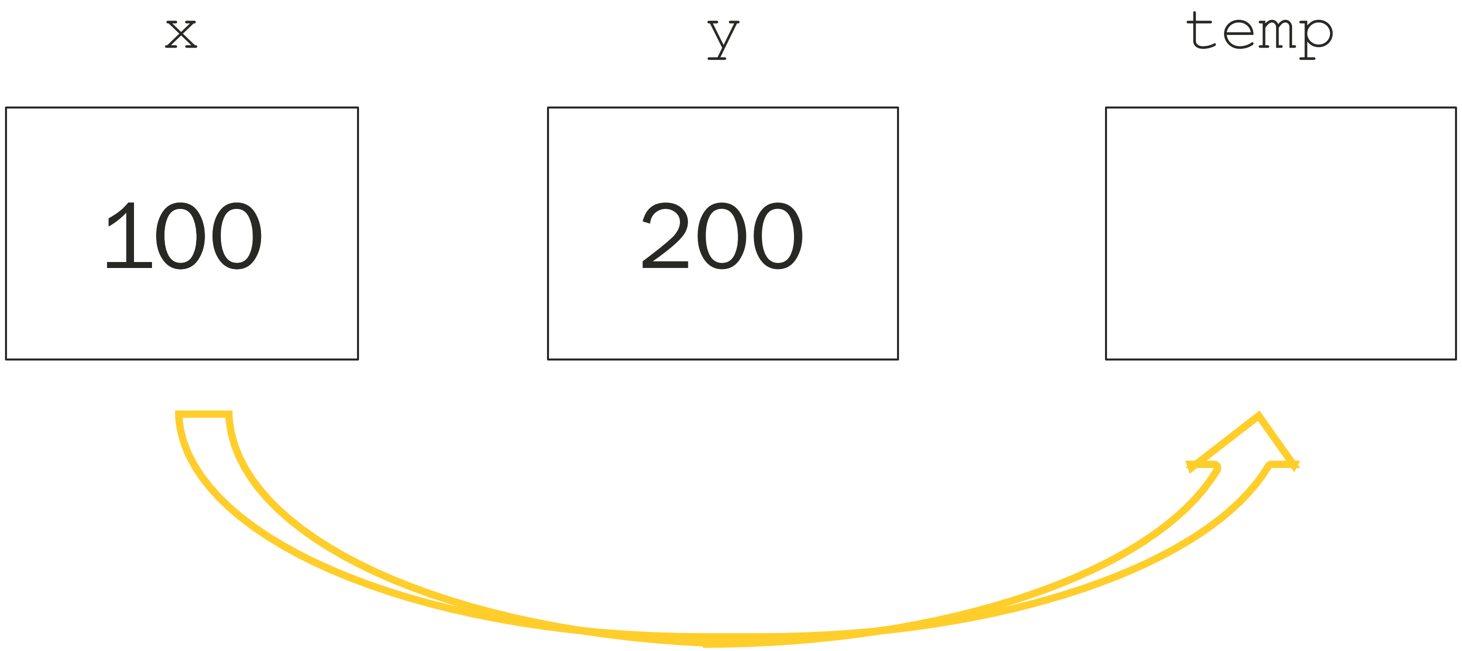
y

200

temp

Before we do any actual data movements

`temp` stores one of the original value, in this case `x`



x

y

temp

100

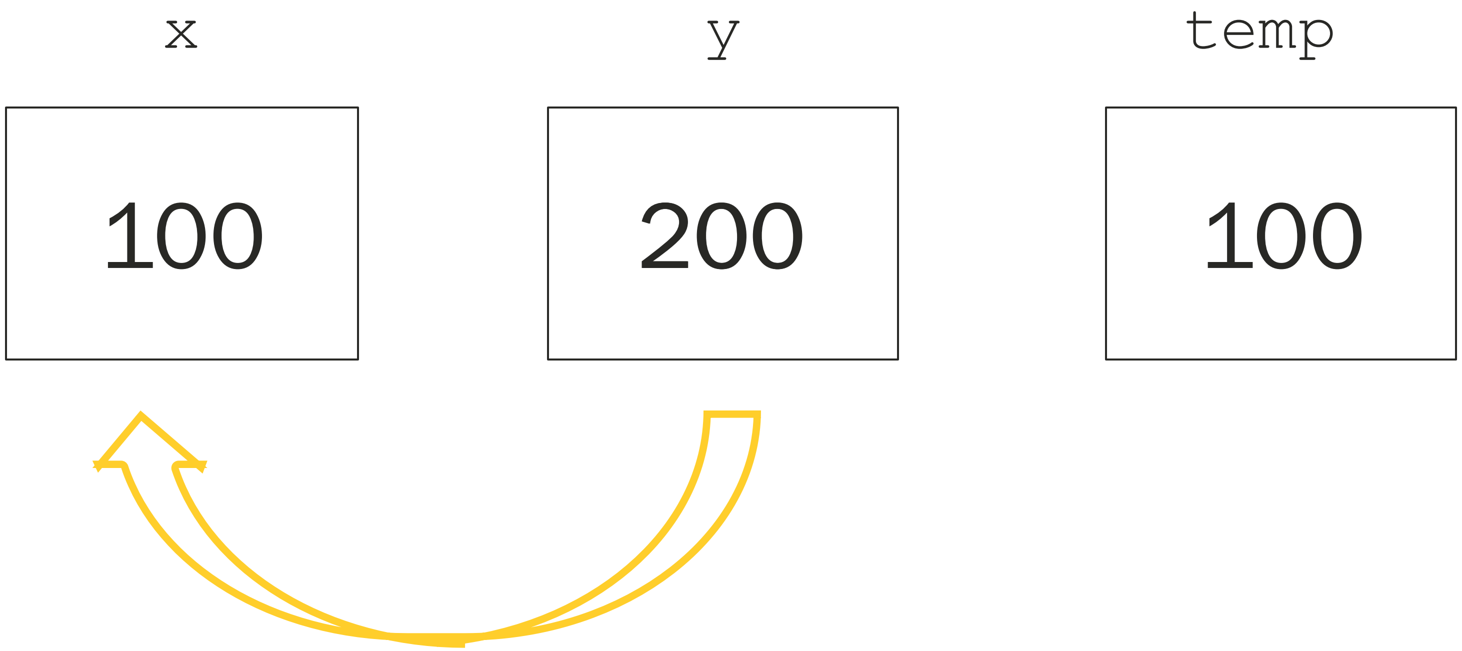
200

Program Codes

```
temp = x;
```

Next, we move data from `y` to `x`

`x` will be replaced by the value of `y`

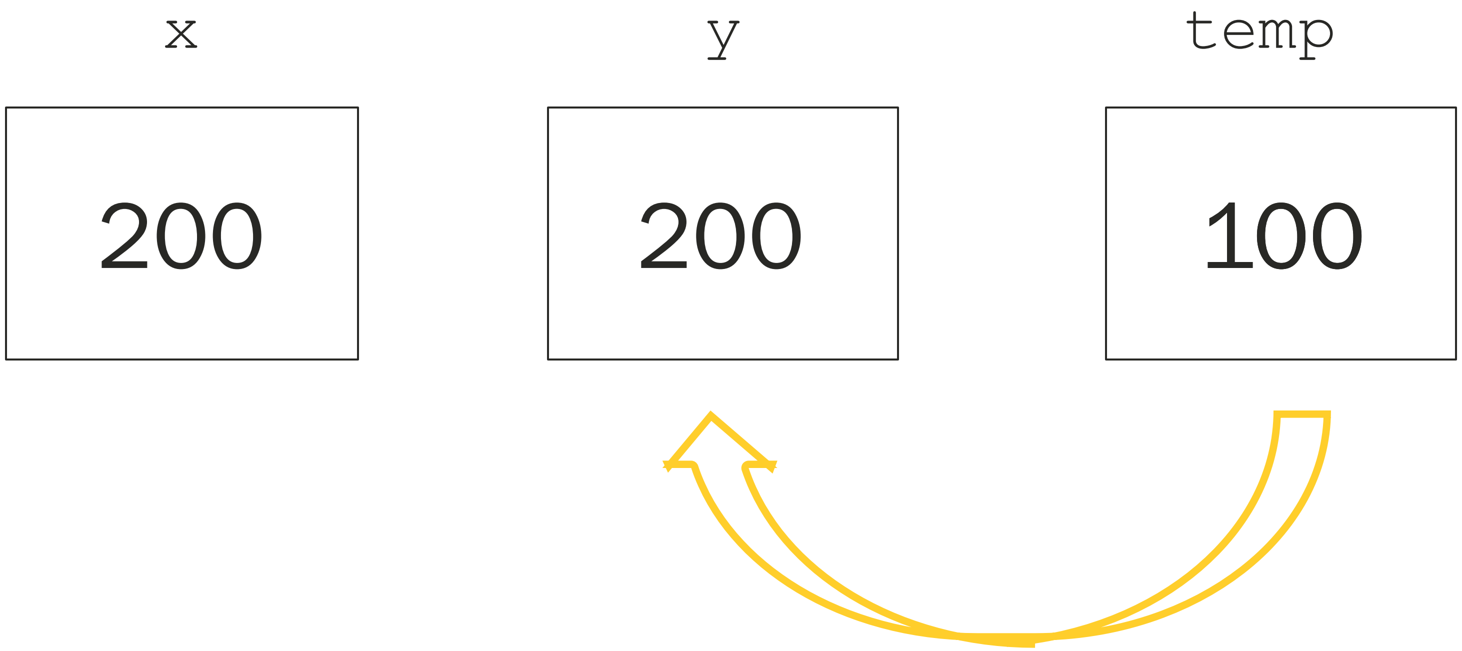


Program Codes

```
temp = x;
x = y;
```

Last, we move data from `temp` to `y`

`y` will be replaced by the value of `temp`

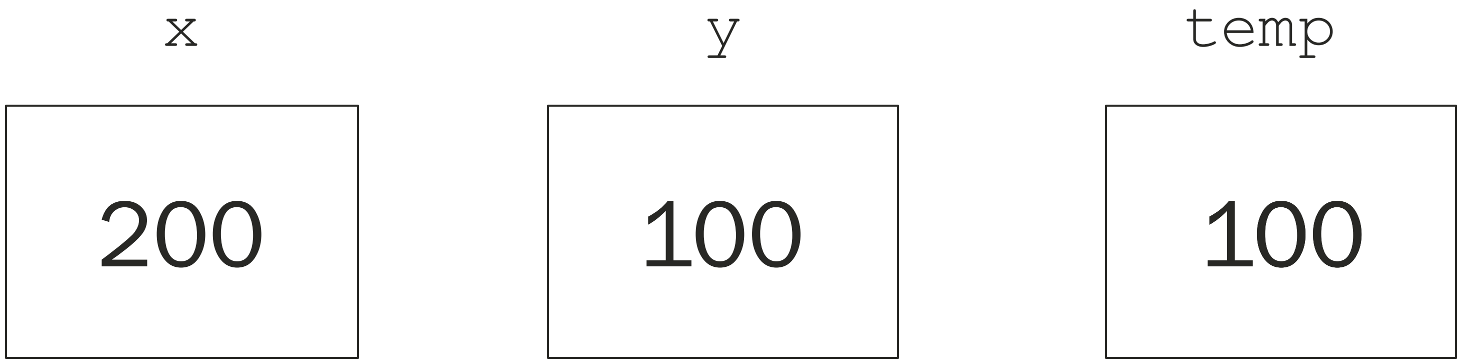


Program Codes

```
temp = x;
x = y;
y = temp;
```

Last, we move data from `temp` to `y`

`y` will be replaced by the value of `temp`



x	y	temp
200	100	100

Program Codes

```
temp = x;
x = y;
y = temp;
```

To Future Data Scientist

- **Bad news** is computer algorithms are difficult to understand, especially low-level programming like C programming

- **Good news** is modern high-level programming language like JavaScript and Python are succinct and much easier to read and code. And many useful algorithms are already built-in.
  - Still, it’s better that you understand what basic computation operations that a computer algorithms are performing

Find the largest number

When there is only a handful of numbers

A human just take a glance and immediately know the answer

10 50 30 60 90

Find the largest number

When there is only a handful of numbers

A human just take a glance and immediately know the answer

10 50 30 60 90

When the numbers grow?

15	12	30	60	20	65
20	53	30	43	17	45
24	93	78	80	92	66
90	34	56	74	78	87
24	88	78	80	92	63

We need a systematic procedure
and that is called an algorithm

When the numbers grow?

15	12	30	60	20	65
20	53	30	43	17	45
24	93	78	80	92	66
90	34	56	74	78	87
24	88	78	80	92	63

We temporary make the first number the `max` (`max = 15`)

Each of the resting number will be compared to `max`, if it is a bigger number, it will be the new `max`

Let’s see the codes

After all, an algorithm is a long set of very simple operations

Operation/Instruction	It’s processing	Code Examples
Conditional flow control	Do processing conditionally	`if () {} else {}`

Comparison Operators	Compare two value. It evaluates to TRUE or FALSE	`if (a==b) {}`
		`if (a!=b) {}`

Logical Operators	Combine multiple logical operations	`if (x>=y && x>=z) {}`

Assignment Operator	Temporary save the value to memory for further processing	`x = 100`
		`i = i + 1`

Looping	Respectively executing the processing	`for (i=0;i<n;i++) {}`
Arithmetic Operators	Mathematic operations	`i = j * 2 - 3`

Simple Search Algorithm

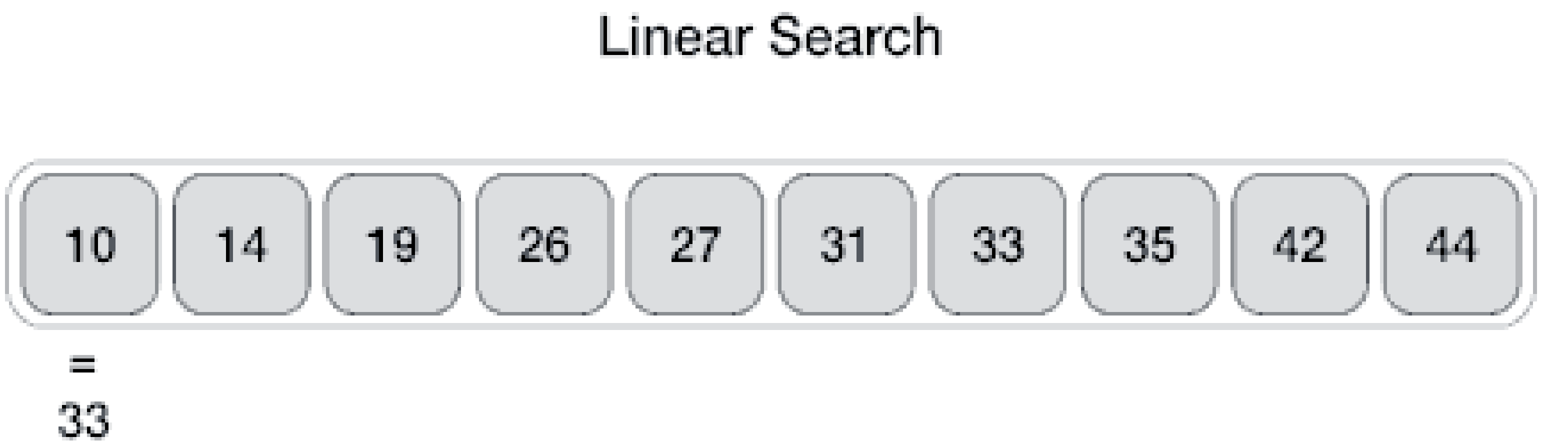
Linear Search

<image/nation>

Visualizing Linear Search

Searching for 33

- Match number by number. Easy to understand.
- Easy to implemented by computer programming.
- Not very effective. Imagine the dataset size nowadays.



Linear Search

10	14	19	26	27	31	33	35	42	44

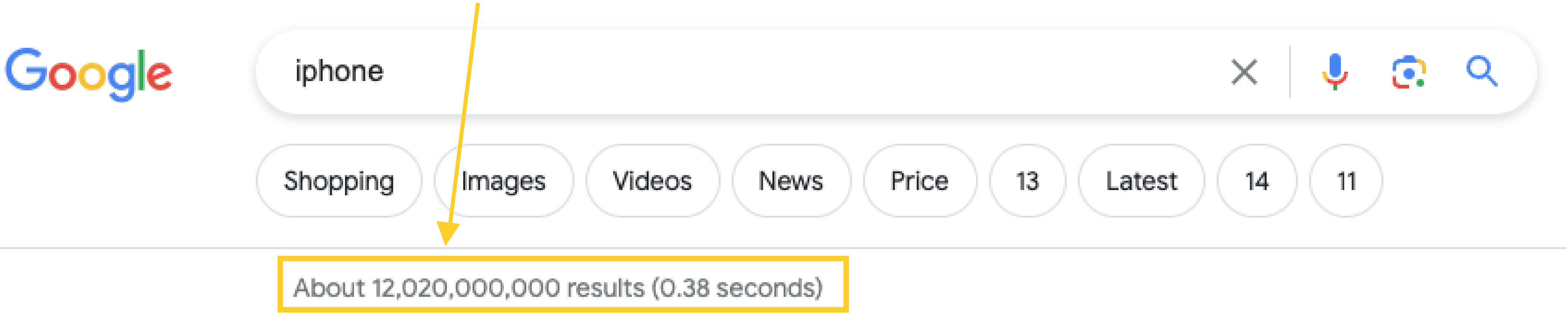
=
33

Linear Search in JavaScript Implementation

```
function search(arr, x) {
  for (i = 0; i < arr.length; i++) {
    if (arr[i] == x) return i; // found
  }
  return -1; // not found
}
```

Linear Search is useful only when the data size is small

Remember the **data size** of Google search is huge?



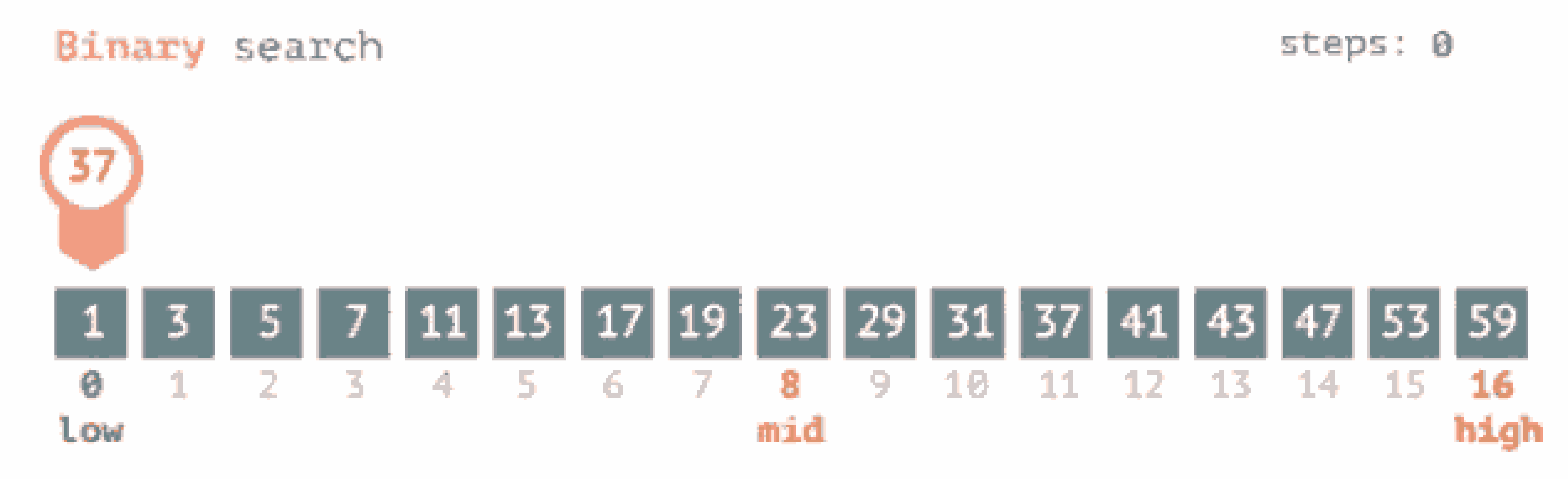
iphone

About 12,020,000,000 results (0.38 seconds)

We need **quicker** search algorithm …

- Turns out if we use certain programming data structures (e.g. **array**) together with special algorithm (e.g. **binary sort**), the searching can be a lot faster.
- But there is a prerequisite
  - The array of data have to be pre-processed - sorted
  - This is the computation cost to faster sorting

We are **halving** the data size in each round of binary search



1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
low								mid								high

The required pre-processing

Sorting the Array

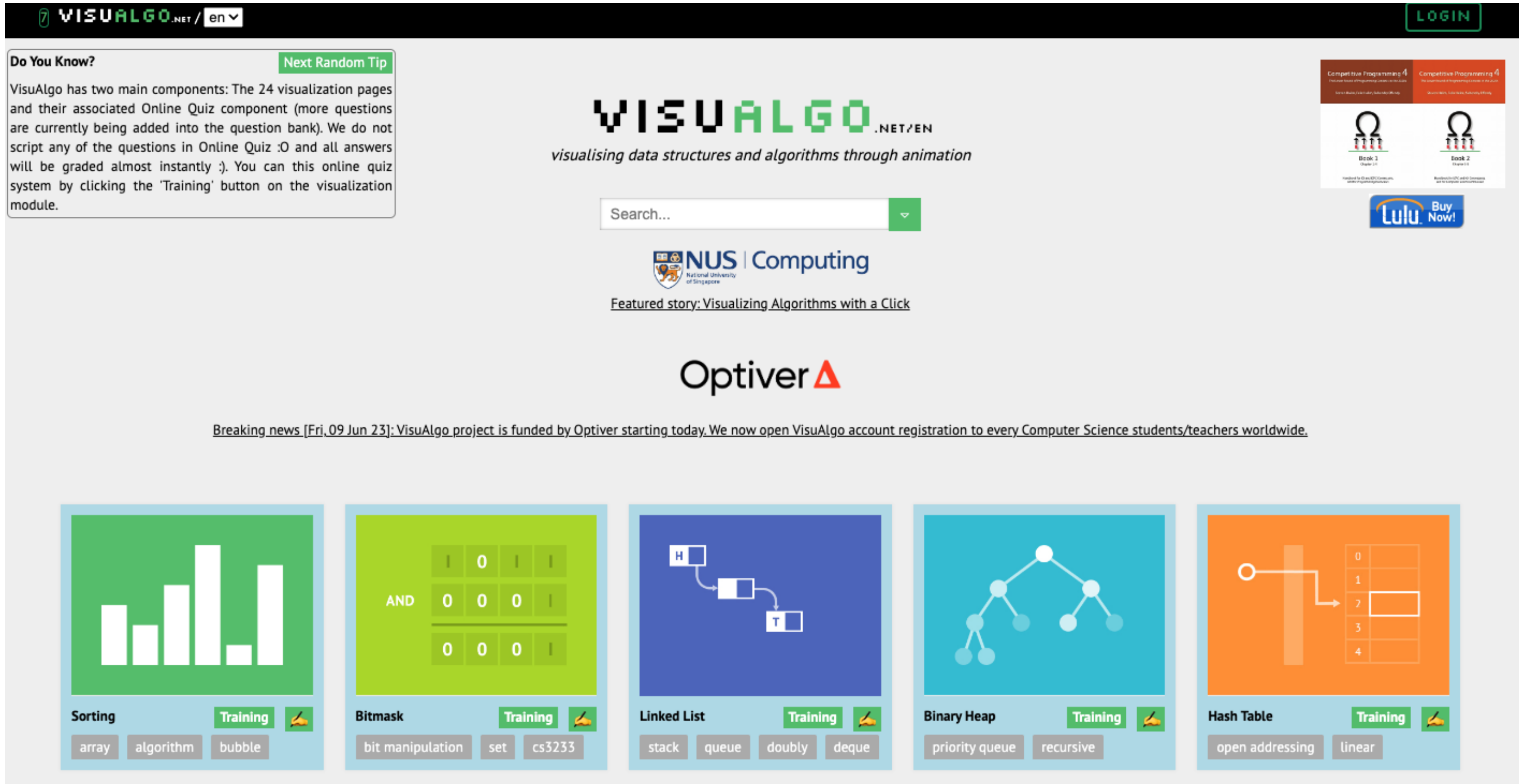
Algorithm Visualization

Help you understand algorithm better

<image/nation>

https://visualgo.net/en

Help you understand algorithm better



Do You Know? Next Random Tip

VisuAlgo has two main components: The 24 visualization pages and their associated Online Quiz component (more questions are currently being added into the question bank). We do not script any of the questions in Online Quiz :O and all answers will be graded almost instantly :). You can this online quiz system by clicking the 'Training' button on the visualization module.

VISUALGO.NET/EN

visualising data structures and algorithms through animation

Search...

NUS | Computing

Featured story: Visualizing Algorithms with a Click

Optiver

Breaking news [Fri, 09 Jun 23]: VisuAlgo project is funded by Optiver starting today. We now open VisuAlgo account registration to every Computer Science students/teachers worldwide.

Computation Complexity

Measuring how efficient an algorithm is

Ultimate Goals of Computation Complexity

- **Time Complexity**
  - Running time or the execution time of operations of data structure must be as small as possible.

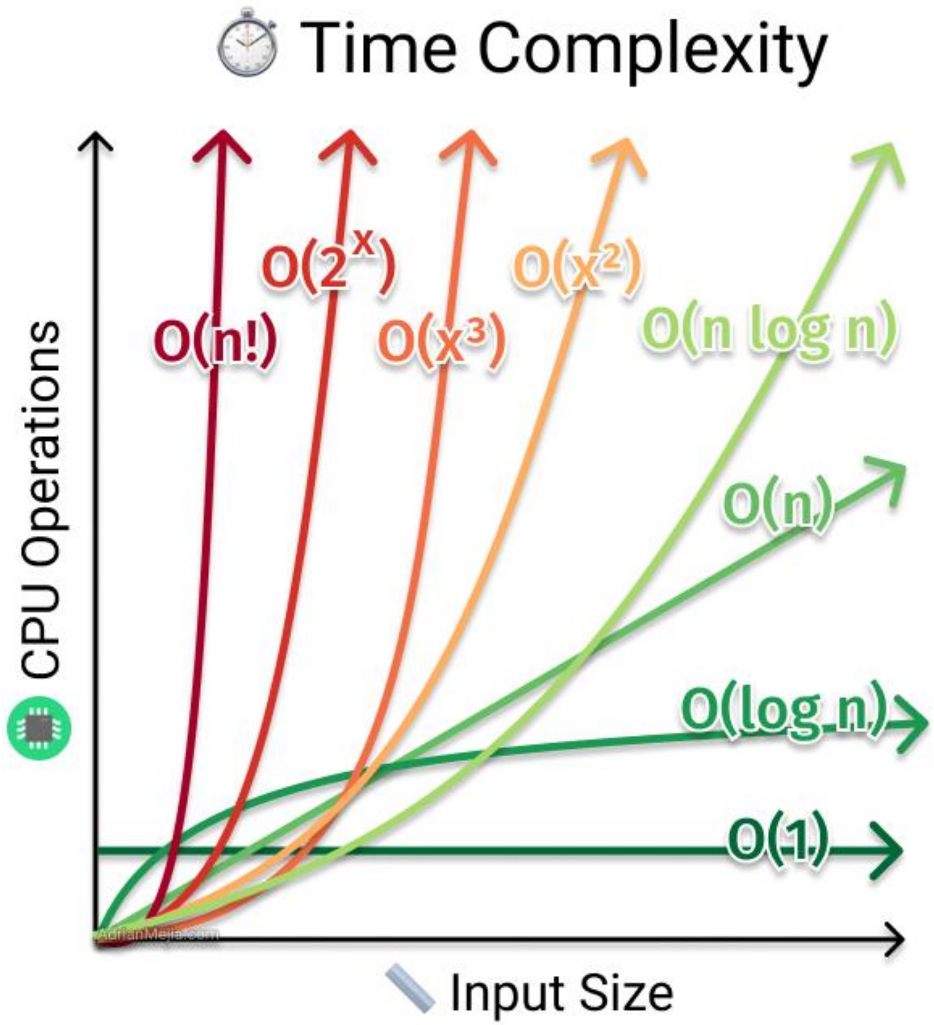
- **Space Complexity**
  - Memory usage of a data structure operation should be as little as possible.

Big-O

Common Time Complexity Measurement

Big-O judges an algorithm by its worst-case performance

E.g. In linear search, it must visit each element at least once and therefore the time complexity is **O(n)**



Time Complexity Examples

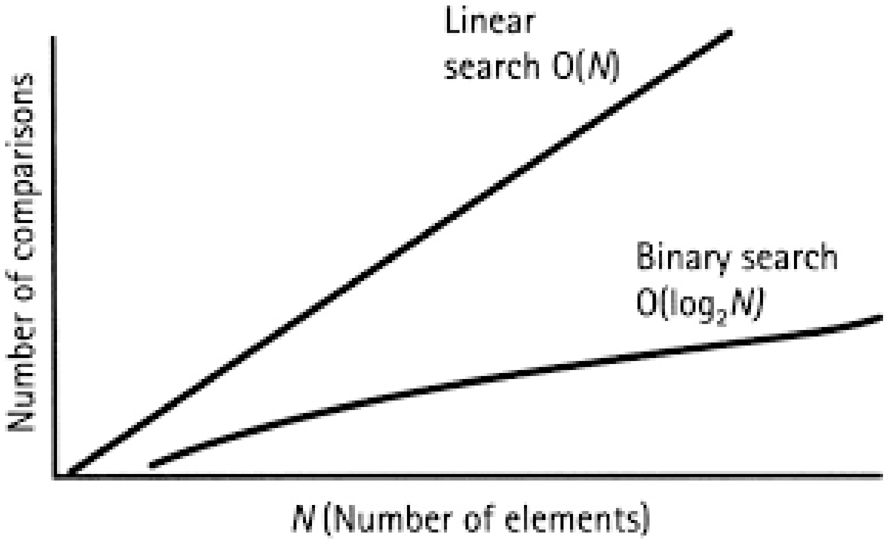
Linear Search

- O(n)
- n is the size of data to search

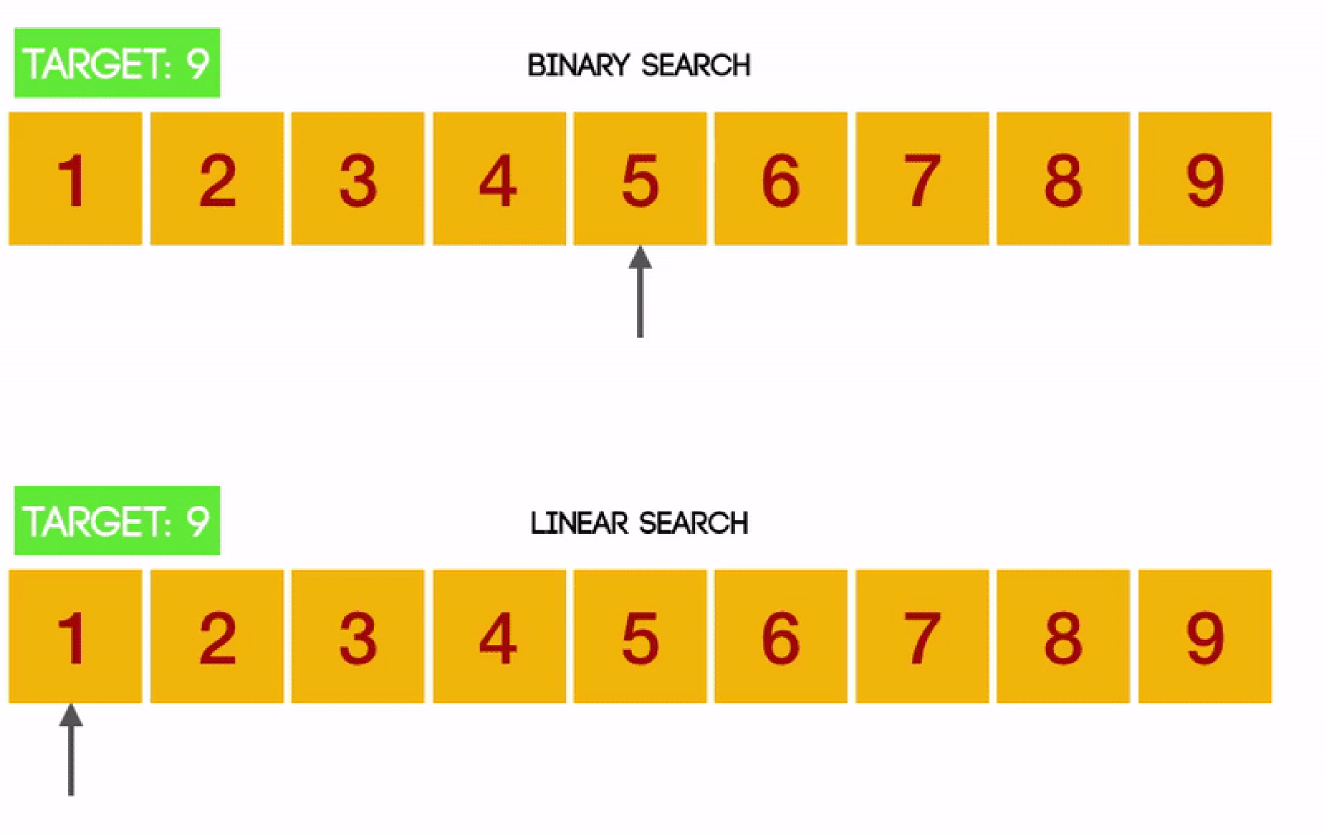
Binary Search

- O(log n)
- n is the size of data to search

Binary Search is a better algorithm for searching



Linear Search vs. Binary Search



When the data size is huge …

If you are searching a number towards **ONE MILLION** numbers

- for linear search, it has to perform **1 million** comparison before you get the answer
- for binary search, it takes at most **20 rounds** to find the target value.

- `log(1,000,000) ~= 20`
- In calculator, you get the result by doing `ln(1,000,000) / ln(2)`

Common data processing algorithm

- **Search** - Algorithm to search an item in a data structure.
- **Sort** - Algorithm to sort items in a certain order.
- **Insert** - Algorithm to insert an item in a data structure.
- **Update** - Algorithm to update an existing item in a data structure.
- **Delete** - Algorithm to delete an existing item from a data structure.

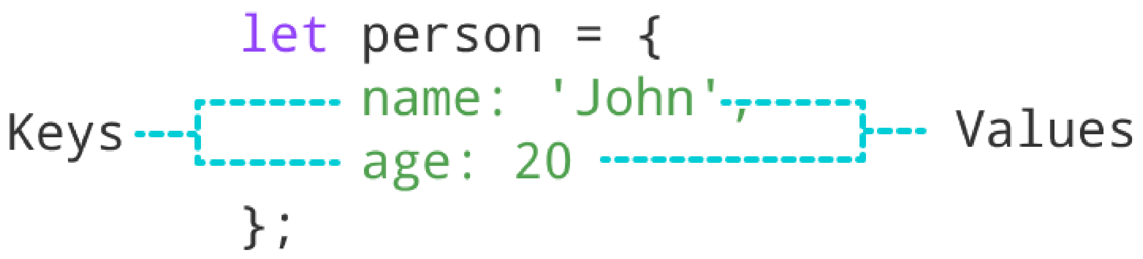
There is NO such thing as the Best Algorithm

- Some data-structure/algorithms are outstanding in time complexity, but NOT GOOD in space complexity
- Some data-structure/algorithms are the other way around
- Some data-structure/algorithms are good in search, but perform poorly in insert, delete and update while some data-structure/algorithms the other way around
- That’s why you need to pick one that suits your use-case

JavaScript Object

JavaScript Object

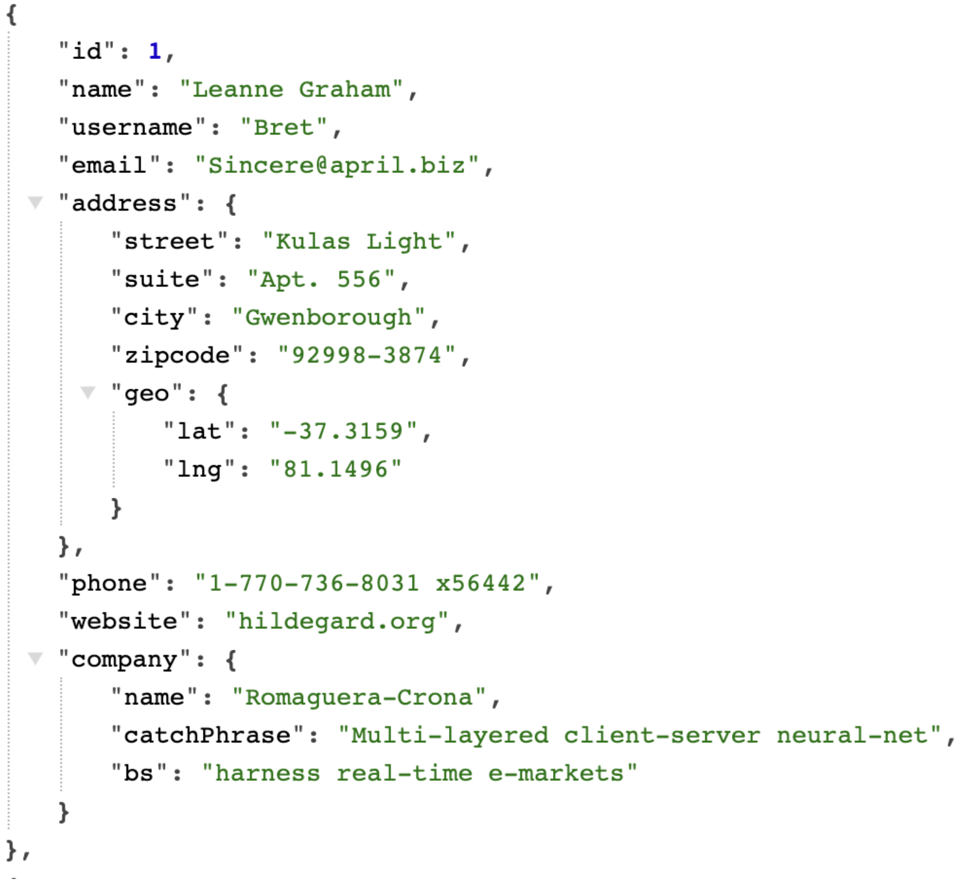
- JavaScript is a compound data structure
- It uses a pair of braces { } to denote object
- In an object, there are multiple pairs of Key-Value.
- Keys and values are separated by a colon : (NOT equals sign =)
  - The key is the attribute name (like a column name to a table)
  - The value could be type of number, string or other
- Each pair of key-value are separated by comas ,



```
let person = {
    name: 'John',
    age: 20
};
```

JavaScript Object can be nested

- JavaScript object is multiple level
- Meaning object can be inside object (and this is quite common)

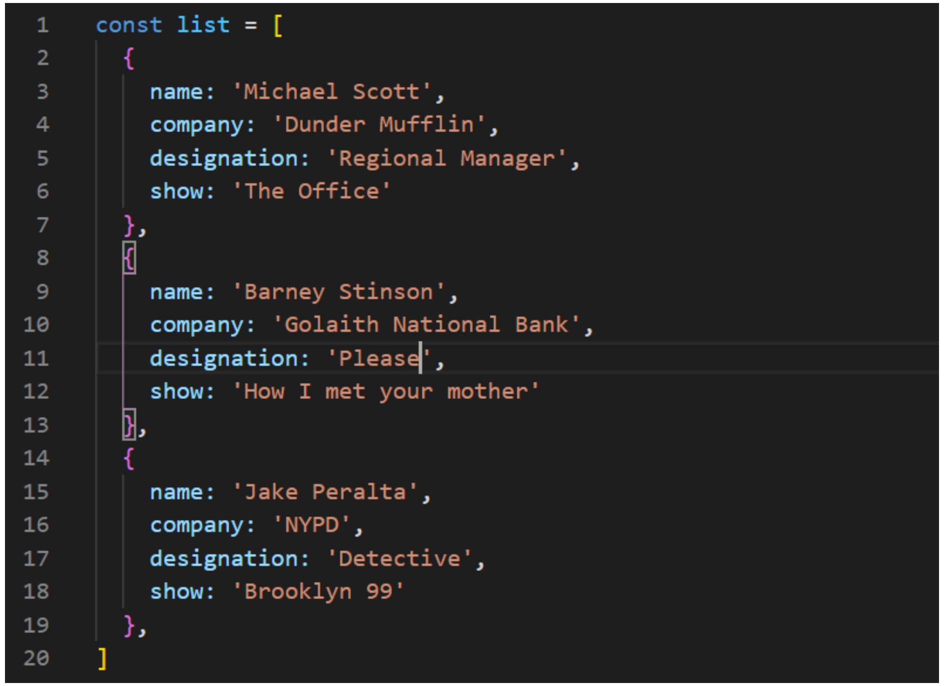


```
{
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
  ▼ "address": {
        "street": "Kulas Light",
        "suite": "Apt. 556",
        "city": "Gwenborough",
        "zipcode": "92998-3874",
      ▼ "geo": {
            "lat": "-37.3159",
            "lng": "81.1496"
        }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
  ▼ "company": {
        "name": "Romaguera-Crona",
        "catchPhrase": "Multi-layered client-server neural-net",
        "bs": "harness real-time e-markets"
    }
},
```

<image/nation>

Multiple objects with same structure form an array/list

- It use a pair of brackets [ ] to denote an array
- Inside the array, there are multiple objects
- Each object is wrapped inside a pair of braces { }
- Objects are separated by a comas ,



```
const list = [
  {
    name: 'Michael Scott',
    company: 'Dunder Mufflin',
    designation: 'Regional Manager',
    show: 'The Office'
  },
  {
    name: 'Barney Stinson',
    company: 'Golaith National Bank',
    designation: 'Please',
    show: 'How I met your mother'
  },
  {
    name: 'Jake Peralta',
    company: 'NYPD',
    designation: 'Detective',
    show: 'Brooklyn 99'
  },
]
```

Hands-on High-Level Language Practicing

2 hours

Source Codes Download

bit.ly/dsa-with-js

Thank You!