We'll be starting shortly!

To help us run the workshop smoothly, please kindly:

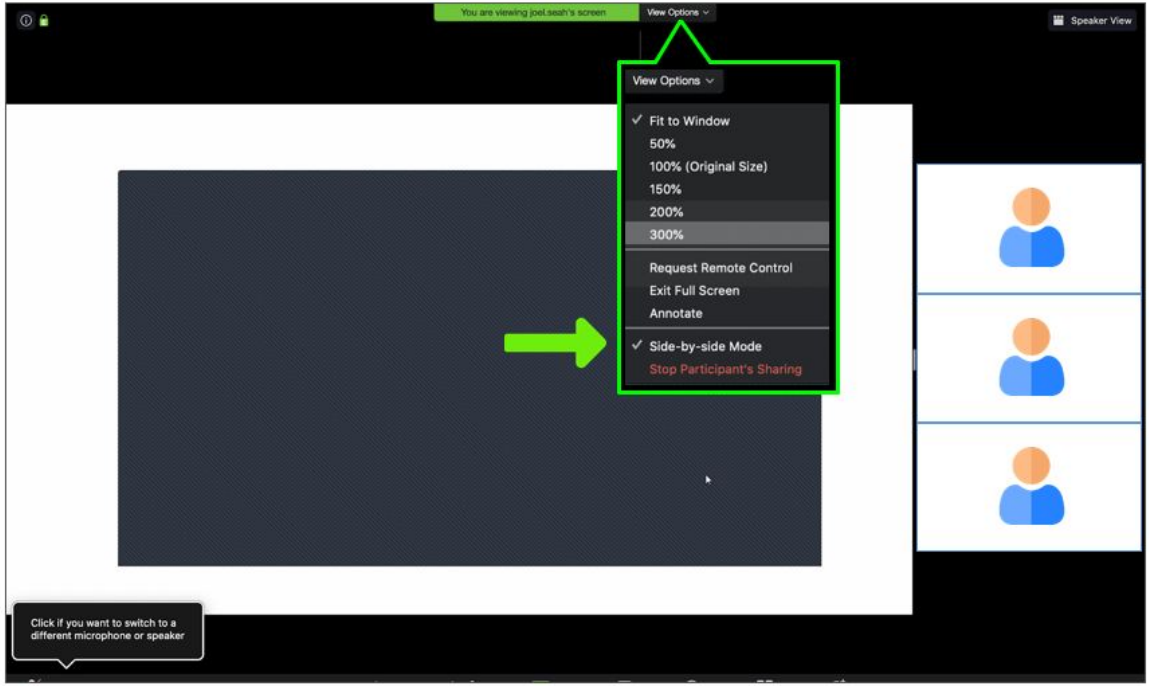
- Submit all questions using the Q&A function
- If you have an urgent request, please use the "Raise Hand" function

Thank you!

Rocket Academy

Shopee CODE LEAGUE 2020

Using Zoom: People & Slides



Side-By-Side Mode

- When sharing screen (slide share)
- With small thumbnails of people on the sidebar

STEPS:

1. View Options
2. Side-By-Side Mode

Rocket Academy

Shopee CODE LEAGUE 2020

Intro to Common Algorithms

Lionell Loh, **Rocket Academy**

Slides: https://tinyurl.com/y7nnxe6n
Register: https://rocketacademy.co/scl-slides

Rocket Academy

Agenda

- **A brief overview of Data Structure and Algorithms**
  - Time Complexity Analysis
- **Hash Table**
  - Some operations
  - Example
- **Search**
  - Binary Search
  - Code Example
  - Binary Search Tree
- **Sorting**
  - Merge Sort
- **Questions and Answer**

Rocket Academy

Shopee CODE LEAGUE 2020

Intended Audience

- You have some basic programming experience
- You have little to no experience with Algorithms
- You are excited to learn!

Rocket Academy

Shopee CODE LEAGUE 2020

Intended Outcomes

- Get a brief overview of Algorithms and Data Structures
- Understand how you can use certain Algorithms and Data Structures to solve certain problems
- Give you a foundation and direction to further your learning

Rocket Academy

Shopee CODE LEAGUE 2020

Before we start

- Sorry in advance if I go too fast or too slowly
- If you have questions, feel free to ask on the Q & A section. I will reserve 5 - 10 mins at the end to answer them. There are no such things as dumb questions!

Rocket Academy

Shopee CODE LEAGUE 2020

What is an Algorithm?

A finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems

Some Examples:

1. Fastest way to get from **Woodlands** to **Upper Changi MRT**

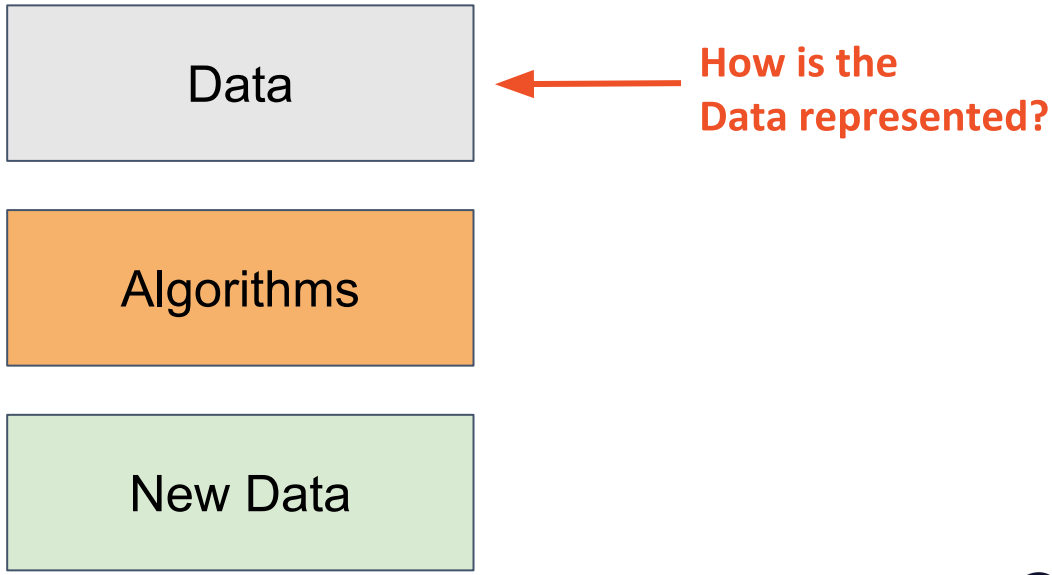
2. Degree to separation between **Bob** and and **Jane** on Facebook

3. All possible English words I can form with the letters “**ROCKET**”

Rocket Academy

Shopee CODE LEAGUE 2020

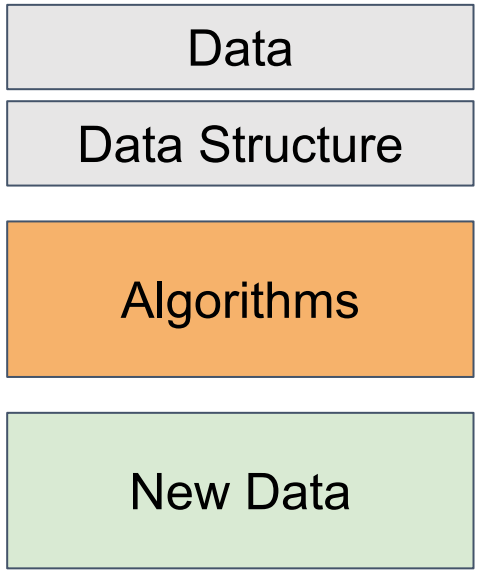
Overview



Rocket Academy

Shopee CODE LEAGUE 2020

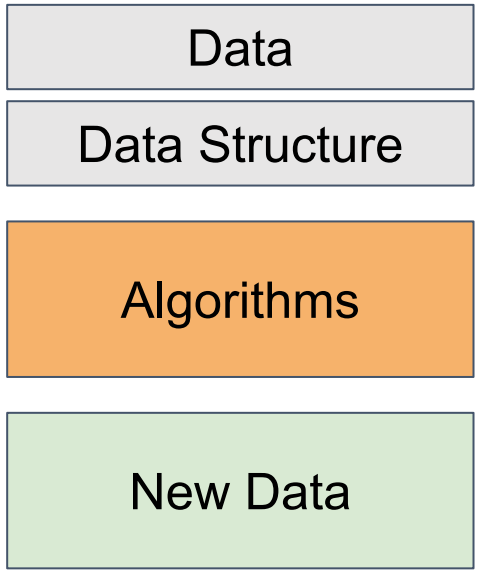
Overview



Rocket Academy

Shopee CODE LEAGUE 2020

Overview



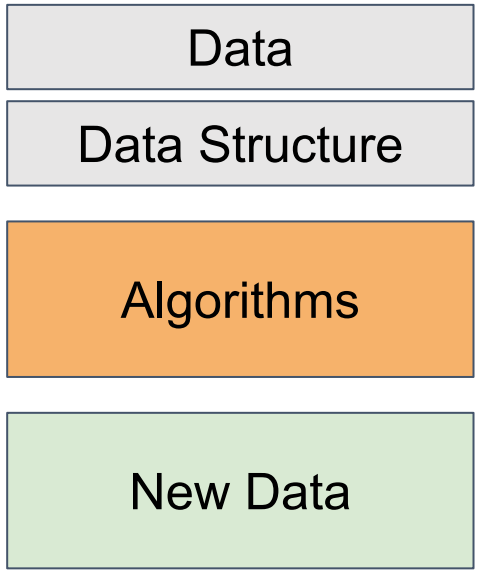
Example Data Structures

- Strings
- Arrays
- LinkedList
- HashTable
- Binary Search Tree
- Heaps
- Merkel Trees

Rocket Academy

Shopee CODE LEAGUE 2020

Overview



Example Data Structures

- Strings
- Arrays
- LinkedList
- **HashTable**
- **Binary Search Tree**
- Heaps
- Merkel Trees

Rocket Academy

Shopee CODE LEAGUE 2020

What is an Algorithm?

An Algorithm can be…

as **simple** as looking through a list of people and figuring out who is the oldest.

as **complex** as recognising sentence structures in language.

Rocket Academy

All roads lead to Rome...

Just as all roads lead to Rome, there can be multiple Algorithms available for a task. How do we choose amongst Algorithms that claim to be able to do something?

1. Correctness
2. Efficiency

Usually most deterministic Algorithms are designed to be correct, and can guarantee correctness

Rocket Academy

Shopee CODE LEAGUE 2020

That leaves us to Efficiency (Time Complexity)

Time Complexity:

Quantifies the amount of time / steps it takes for an Algorithm to finish running.

When we calculate the time complexity of an Algorithm, we are trying to find out how efficient it is. **Often with respect to the size of the data**. **We also often assume the worst case scenario.**

To motivate this further: Designing an Algorithm with a good time complexity during a technical interview can help you land a job. :-)

Rocket Academy

Time Complexity

Imagine there is a group of 10 people -

You as the newcomer need to have talk to them on a personal basis

How many times do you have to introduce yourself?

Questions:

1. What happens if there are 100 people? Or just 3 people?
2. What happens if there are **N** people?
3. What is the Time Complexity?

O(N)

Rocket Academy

Time Complexity

Imagine there is a group of 10 people -

You as the newcomer need to arrange two 1-on-1 sessions with each person.

How many sessions do you have to go for?

Questions:

1. What happens if there are 100 people? Or just 3 people?
2. What happens if there are **N** people?
3. What is the time complexity?

$$O(2N) = O(N)$$

Rocket Academy

Shopee CODE LEAGUE 2020

Time Complexity

Imagine there is a group of 10 people -

You as the newcomer gather everyone and introduce yourself at one go.

How many times do you have to introduce yourself?

Questions:

1. What happens if there are 100 people? Or just 3 people?
2. What happens if there are **N** people?
3. What is the time complexity?

O(1)

Rocket Academy

Time Complexity

Imagine there is a group of 10 newcomers

Everyone has to introduce themselves on 1 - 1 basis, but there can only be one pair introducing themselves to each other at any one time.

Questions:

1. What happens if there are 100 people? Or just 3 people?
2. What happens if there are n people?
3. What is the time complexity?

$$(n-1) + (n-2) + \ldots + 2 + 1 = (0.5)\,(n-1)\,(n) = 0.5n^2 - 0.5n$$

1. Only retain the dominant term
2. Drop all constants

$$O(n^2)$$

Rocket Academy

Code Examples

```
def f0(x: list):
    for i in range(len(x)):
        print(x[i])

    for j in range(len(x)):
        print(x[j)
```

O(N)

Code Examples

```
def f1(x: list):
    for i in range(len(x)):
        for j in range(i, len(x)):
            print(x[i], x[j])
```

$O(N^2)$

Rocket Academy

Code Examples

```
def f2(x: list):
    for i in range(len(x)):
        for j in range(i, len(x)):
            for k in range(j, len(x)):
                print(x[i], x[j], x[k])
```

$O(N^3)$

Rocket Academy

Time Complexity

Rules to calculating time complexity

- Drop the constants
- Only keep the dominating term

- $2N^3 + 30N^2 = \mathbf{N^3}$
- $N(\log N) + 5N + 16\log N = \mathbf{N(\log N)}$

At the end of the day time complexity is about categorising the algorithm's efficiency as the **size of the data approaches infinity**

Rocket Academy

Shopee CODE LEAGUE 2020

Categories in Time Complexity

Efficient Algorithms **Infeasible Algorithms**

O(1)	O(log(log(N))	O(log(N))	O(N)	O(N log(N))	$O(N^c)$	$O(c^N)$	O(N!)	$O(N^N)$

Rocket Academy

Shopee CODE LEAGUE 2020

Hash Table

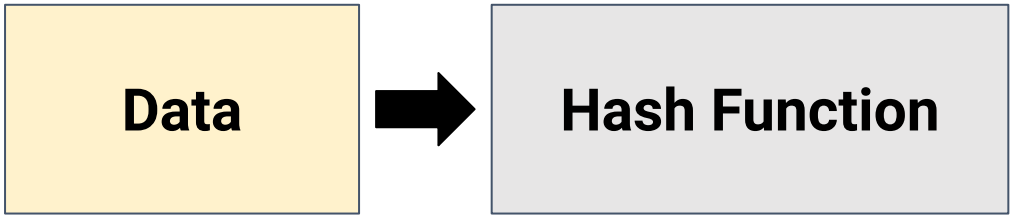
1. A data structure that maps keys to values
2. **Constant Time - O(1)** for search, insert, and delete operations

Hash Table in Python

```
ages = {"Bob": 5, "Mary": 13, "John:": 22}

#search/get/retrieve
print(ages["Bob"]) #=> 5

#delete
del ages["Bob"]

#insert/assignment
ages["Aaron"] = 17
```

Rocket Academy

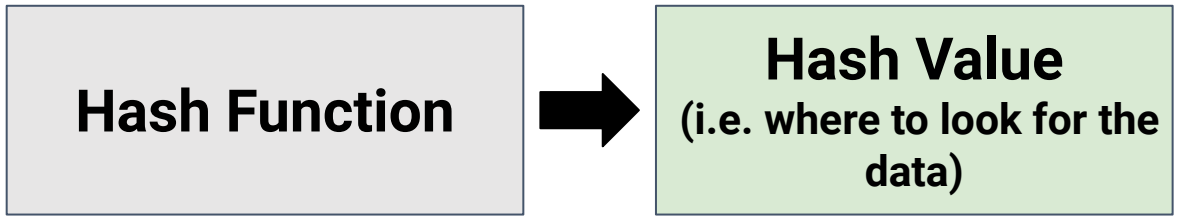
How it works (briefly)



Rocket Academy

Shopee CODE LEAGUE 2020

How it works (briefly)



Rocket Academy

Shopee CODE LEAGUE 2020

How it works (briefly)

Rocket Academy

Shopee CODE LEAGUE 2020

Two Sum Problem

Given an array of integers, return indices of the two numbers such that they add up to a specific target.

*You may assume that each input would have **exactly one solution**, and you may not use the same element twice.*

```
Given nums = [2, 13, 16, 8], target = 10
        We should return [0, 3]
```

Rocket Academy

Two Sum Problem

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

```
for i in range(len(nums)):
  for j in range(i + 1, len(nums)):
    if nums[i] + nums[j] == target:
      return [i, j]
```

$O(N^2)$

We are seeing each element many times!

Rocket Academy

Shopee CODE LEAGUE 2020

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Key	Value

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Key (num)	Value (index)
2	0
13	1
16	2
8	3

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Index = 0
Num = 2
Diff = 8

Is 8 inside?

Key (num)	Value (index)

Rocket Academy

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Index = 1
Num = 13
Diff = -3

Is -3 inside?

Key (num)	Value (index)
2	0

Rocket Academy

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Index = 2
Num = 16
Diff = -6

Is -6 inside?

Key (num)	Value (index)
2	0
13	1

Rocket Academy

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Index = 3
Num = 8
Diff = 2

Is 2 inside?

Key (num)	Value (index)
2	0
13	1
16	2

Rocket Academy

Shopee CODE LEAGUE 2020

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
We should return [0, 3]
```

What if we used a Hash Table to store what we have seen?

Index = 3
Num = 8
Diff = 2

Is 2 inside?

Key (num)	Value (index)
2	0
13	1
16	2

Return [0, 3]

Two Sum Problem (using a Hash Table)

```
Given nums = [2, 13, 16, 8], target = 10
        We should return [0, 3]
```

```
def twoSum(nums, target)
    seen = {}
    for i in range(len(nums)):
        if (target - nums[i]) in seen: # O(1)
            return [seen[target - nums[i]],
        else:
            seen[nums[i]] = i  #O(1)
```

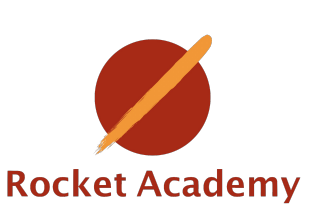
Rocket Academy

Shopee CODE LEAGUE 2020

Caveats with Python's Dictionary

The key of the dictionary cannot be a mutable data type.

If the content of the key is changed, the hash value will change too.



Search and Sort

Search

- In a sea of data, how can find data that I need?
- E.g. Google Search, Searching for items on Shopee

Sort

- Create order from disorder
- Easier to view and analyse sorted data
- Preprocessing for certain Algorithms
- Sorting items on Shopee by price and relevance

Rocket Academy

Shopee CODE LEAGUE 2020

Binary Search

An algorithm that searches a sorted array by repeatedly dividing the search interval into two.

Binary Search

Find the position of number 17

1	18	5	6	20	13	17	21	3	9	11	4	6	10	11	10	21	23	25	30

Rocket Academy

Binary Search

Find the position of number 17

1	18	5	6	20	13	17	21	3	9	11	4	6	10	11	10	21	23	25	30

What if we have a sorted array?

1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

Rocket Academy

Binary Search

Find the position of number 17

1	18	5	6	20	13	17	21	3	9	11	4	6	10	11	10	21	23	25	30

What if we have a sorted array?

1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

Rocket Academy

Binary Search

What if we have a sorted array?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

Rocket Academy

Binary Search

What if we have a sorted array?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

10	11	12	13	14	15	16	17	18	19
11	13	17	18	20	21	23	25	25	30

Rocket Academy

Binary Search

What if we have a sorted array?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

10	11	12	13	14	15	16	17	18	19
11	13	17	18	20	21	23	25	25	30

10	11	12	13
11	13	17	18

Binary Search

What if we have a sorted array?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	5	6	6	9	10	10	11	11	13	17	18	20	21	23	25	25	30

10	11	12	13	14	15	16	17	18	19
11	13	17	18	20	21	23	25	25	30

10	11	12	13
11	13	17	18

12	13
17	18

Binary Search (Code Example)

```
def binarySearch(arr, x):
    l = 0
    r = len(arr) - 1
    while l <= r:
        mid = l + (r - l) // 2;
        if arr[mid] == x:
            return mid
        elif arr[mid] < x:
            l = mid + 1
        else:
            r = mid - 1

    return -1
```

Rocket Academy

Binary Search

How many times must we split the array to eventually reach an array with one element?

Let N be the length of the array.

Let X be the number of times we need to split the array

$N = 2^x$

$\log_2(N) = x\ (\log_2 2)$

$x = \log(N)$

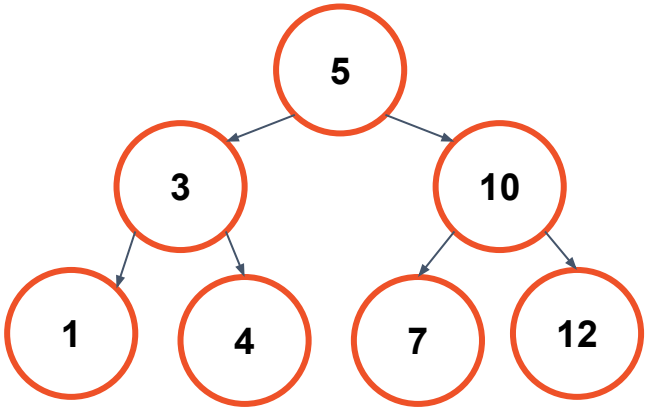
Rocket Academy

Binary Search Tree (BST)

A Binary Search Tree has the following properties

- Each node has two children (See “Binary”!)
- The right subtree of a node only contains nodes with key greater than the node’s key
- The left subtree of a node only contains nodes with keys greater than the node’s key

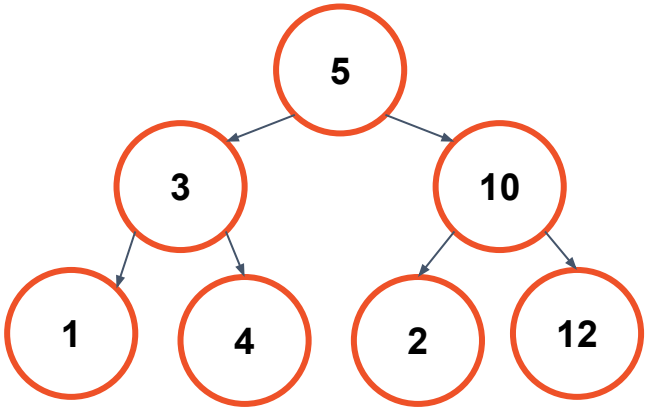
Binary Search Tree (BST)



In a perfect balanced BST, what is the time complexity of search?

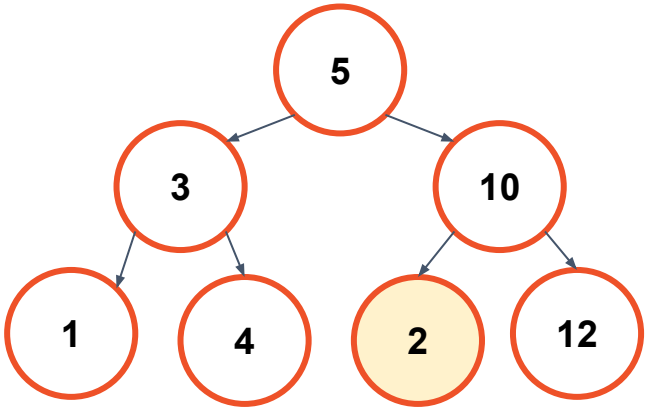
Rocket Academy

Let's Play - Is this a BST?



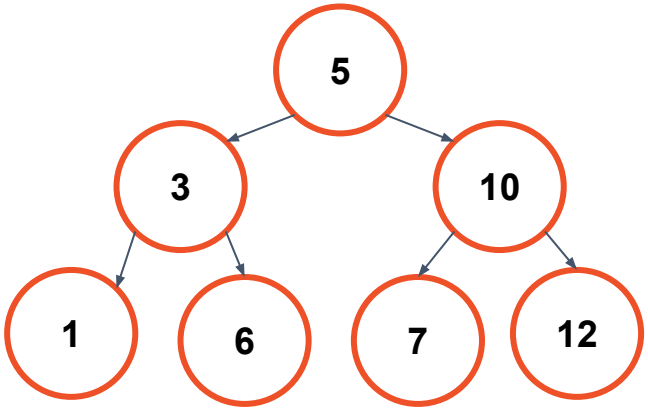
Rocket Academy

Let's Play - Is this a BST?



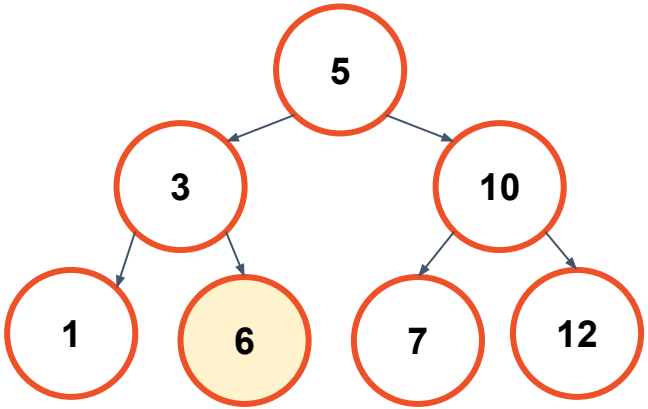
Rocket Academy

Let's Play - Is this a BST?



Rocket Academy

Let's Play - Is this a BST?



Rocket Academy

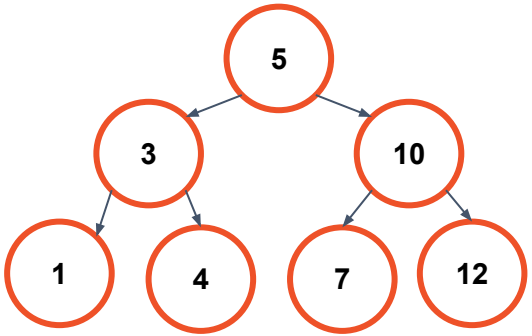
Binary Search Tree

In a perfect balanced BST, what is the time complexity of search?

num_levels	num_nodes (N)
0	1 (+2)
1	3 (+4)
2	7 (+8)
3	15 (+16)
4	31 (+32)
5	63

$$num\_nodes = 2^{num\_levels + 1} - 1$$

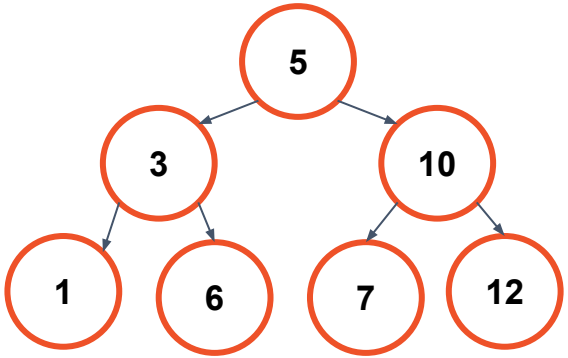
$$num\_levels = \log_2(num\_nodes) = \log(N)$$

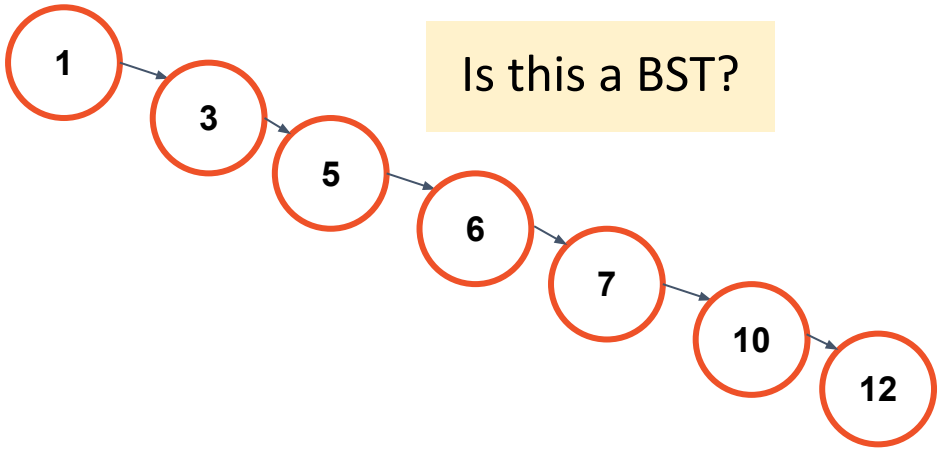


Rocket Academy

Shopee CODE LEAGUE 2020

Binary Search Tree

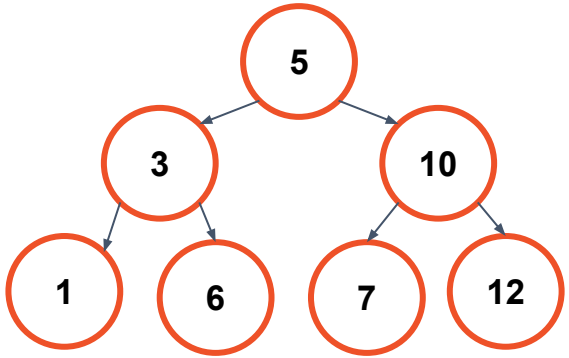


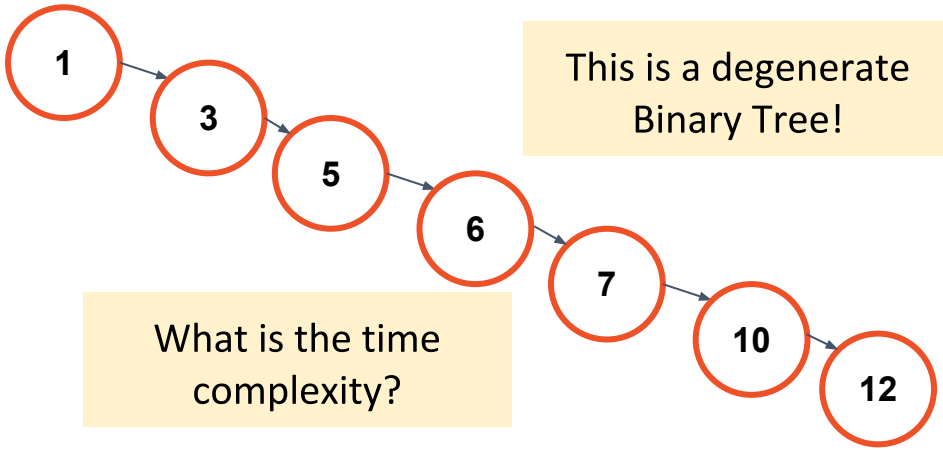


Is this a BST?

Rocket Academy

Binary Search Tree





This is a degenerate Binary Tree!

What is the time complexity?

Caveats

- Must be Almost Balanced or Perfect Balanced Trees
- Self-Balancing Algorithms (Red-Black, AVL Trees)

Rocket Academy

Shopee CODE LEAGUE 2020

Merge Sort

- A out-of-place, stable sorting algorithm
- Uses the Divide and Conquer strategy

Rocket Academy

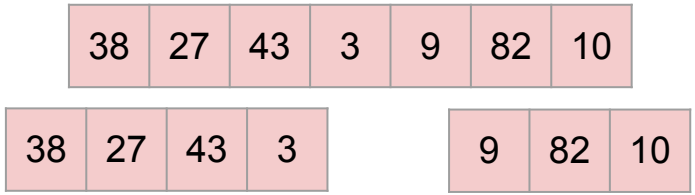
Shopee CODE LEAGUE 2020

MergeSort

38	27	43	3	9	82	10

Rocket Academy

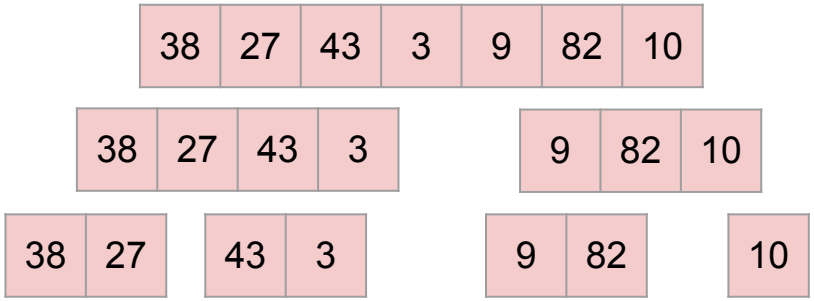
MergeSort



Rocket Academy

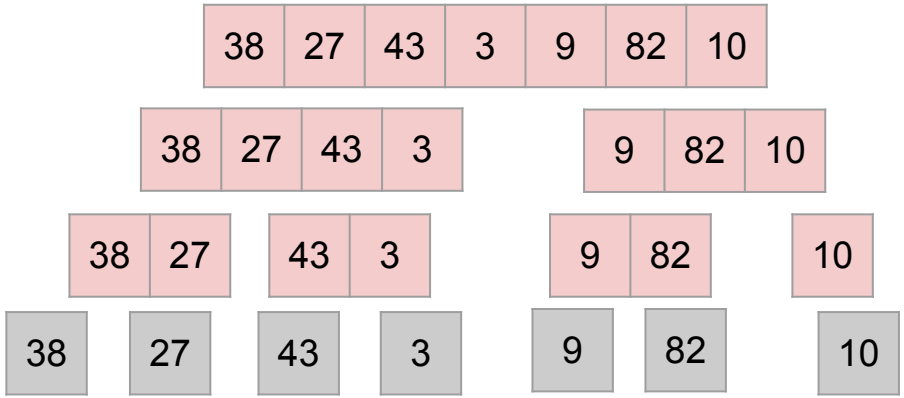
Shopee CODE LEAGUE 2020

MergeSort



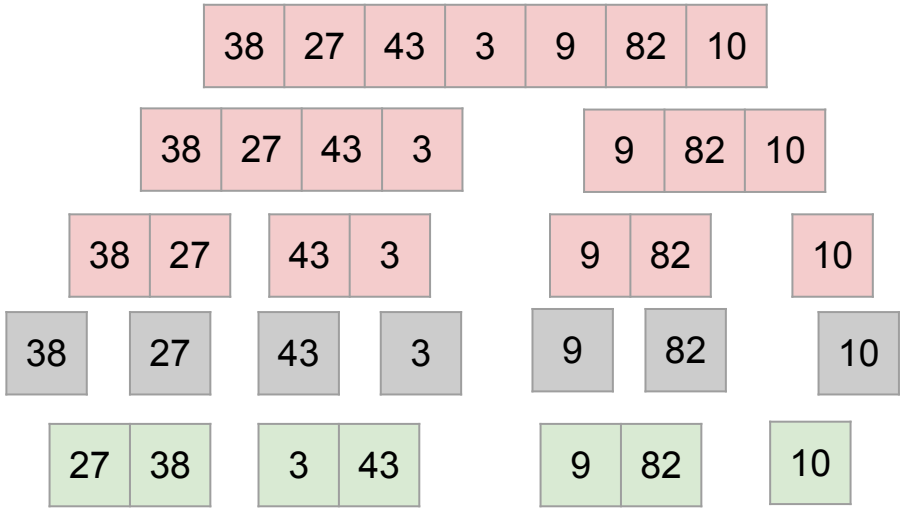
Rocket Academy

MergeSort



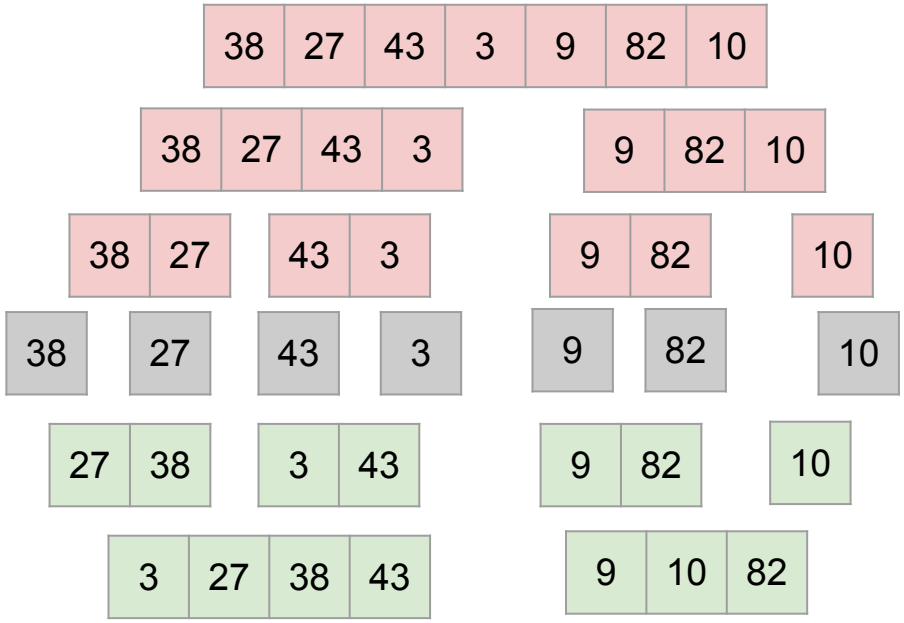
Rocket Academy

MergeSort



Rocket Academy

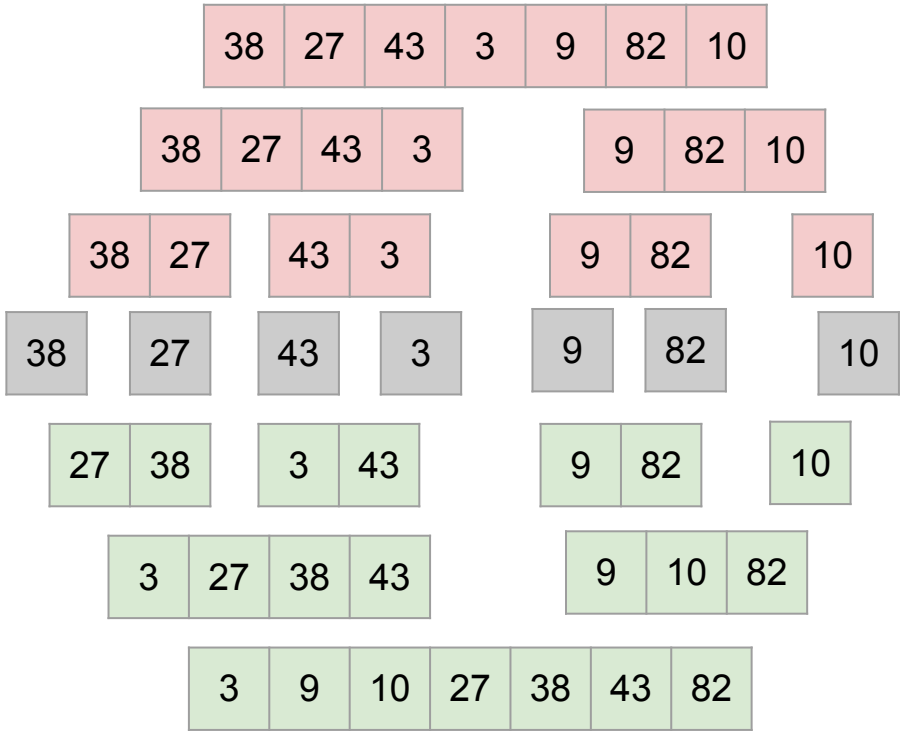
MergeSort



Rocket Academy

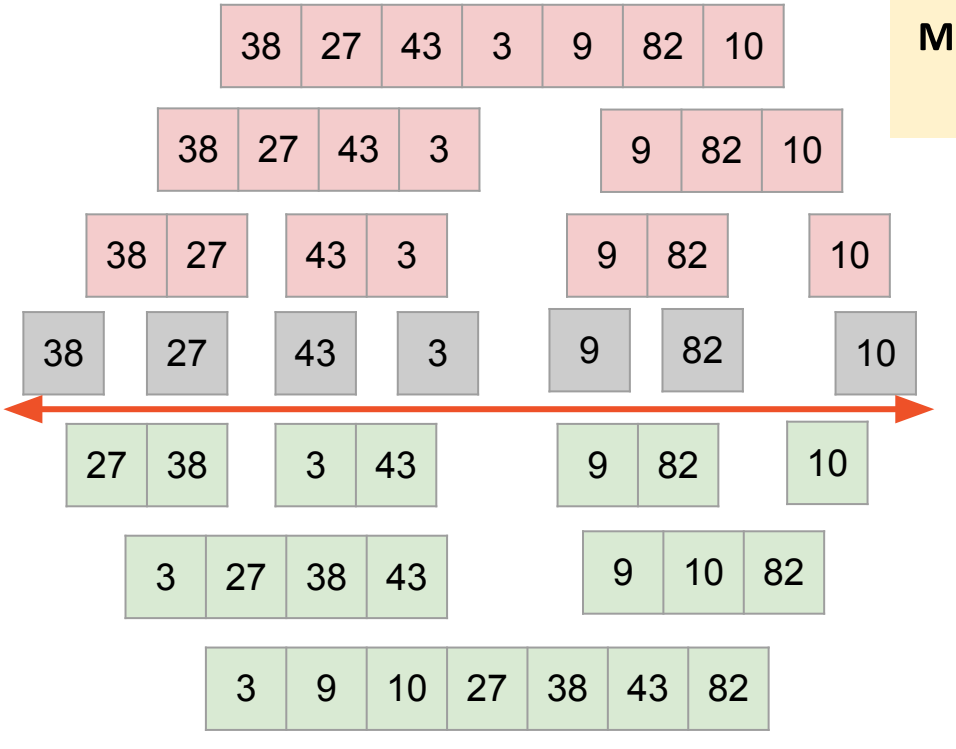
Shopee CODE LEAGUE 2020

MergeSort



Rocket Academy

MergeSort



Height =
O(2 log(N)) =
O (log (N))

Merge Sort Time Complexity:
O(N log N)

Merging 1 layer:
O(N)

Rocket Academy

MergeSort (Code Example)

```
def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2 # Finding the mid of the array
        L = arr[:mid] # Dividing the array elements
        R = arr[mid:] # into 2 halves
        mergeSort(L) # Sorting the first half
        mergeSort(R) # Sorting the second half
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i+= 1
            else:
                arr[k] = R[j]
                j+= 1
            k+= 1

        while i < len(L):
            arr[k] = L[i]
            i+= 1
            k+= 1

        while j < len(R):
            arr[k] = R[j]
            j+= 1
            k+= 1
```

Source: Geeks For Geeks

Rocket Academy

Summary

	Indexing	Search	Insertion	Deletion
Hash Table	-	O(1)*	O(1)*	O(1)*
Binary Search Tree	O(log N)/O(N)	O(log n)/O(N)	O(log N)/O(N)	O(log N)/O(N)
Balanced BST	O(log N)	O(log N)		
Binary Search	O(log N)			
Merge Sort	O(N log N)			

Legend: Avg / Worst Case Time Complexity

*O(1) for hash table is if we assume **(1)** Chaining to resolve hash collisions
(2) Uniform Hash Function and **(3)** Constant Load Factor through Hash Table Resizing

Rocket Academy

Shopee CODE LEAGUE 2020

Language Data Types vs Canonical Data Structures

It is important to know what is happening under the hood

- A **Python** List -> A Linked List or an Array?
- A **Python** Dictionary -> ?
- A **Javascript** Object -> ?
- A **Java** ArrayList -> ?

Rocket Academy

Shopee CODE LEAGUE 2020

Programmatic API vs Algorithms

It is important to know what is happening under the hood

- `"Rocket" + "Academy"`
- `if x in arr: …`
- `max_int = max(arr)`
- `arr.sort()`
- `new_arr = arr[:]`

Rocket Academy

Shopee CODE LEAGUE 2020

Parting Words

1. Skepticism and curiosity are important when learning about Algorithms

2. Algorithmic thinking takes conscious effort

3. Optimizing Algorithms starts to pay off when there is scale

Intro to Common Algorithms

Lionell Loh, **Rocket Academy**

Slides: https://tinyurl.com/y7nnxe6n
Register: https://rocketacademy.co/scl-slides

Rocket Academy

Your Feedback Matters!

https://techatshopee.formstack.com/forms/shopeecodeleague_workshopfeedbackform

Rocket Academy

Shopee CODE LEAGUE 2020

Hash Functions

Hash function: A hash function is any function that can be used to map a data set of an **arbitrary size** to a **data set of a fixed size reproducibly**, which falls into the hash table. The values returned by a hash function are called hash values or hashes.

hash(“Rocket Academy”) -> 6

In this case, **hash()** is the **hash function**.

“Rocket Academy” is the **input.**

6 is the **hash value**

Rocket Academy

Shopee CODE LEAGUE 2020

Hash Table

age[0]	NULL
age[1]	NULL
	NULL
	NULL
	NULL
	NULL
	NULL
age[7]	NULL

hash(“Bob”) -> 5

age[5] = BOB_AGE

hash(“Amy”) -> 3

age[3] = AMY_AGE

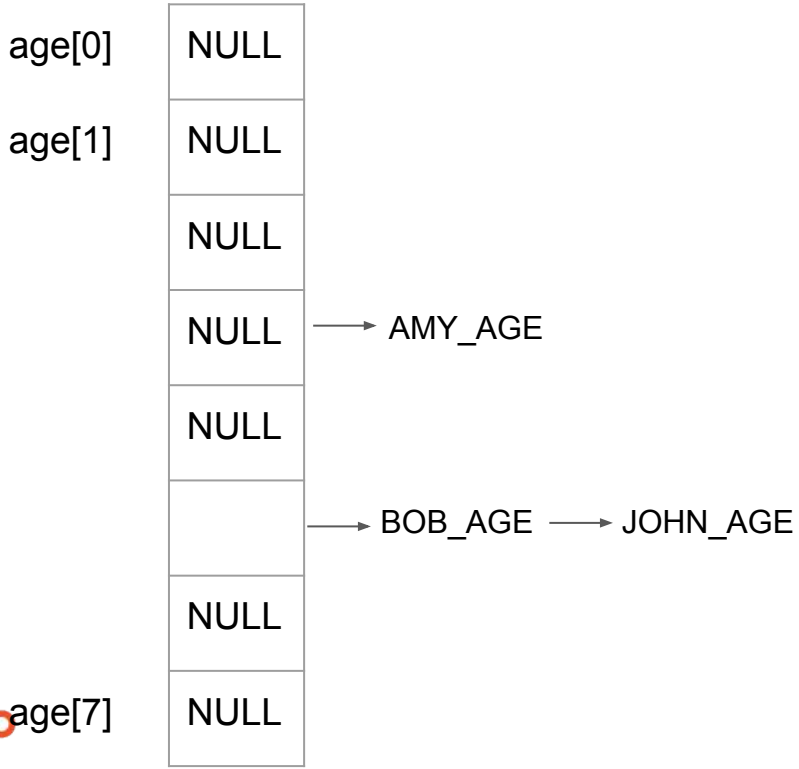
hash(“John”) -> 5

age[5] = JOHN_AGE

Rocket Academy

Shopee CODE LEAGUE 2020

Hash Table



hash(“Bob”) -> 5

age[5] = BOB_AGE

hash(“Amy”) -> 3

age[3] = AMY_AGE

hash(“John”) -> 5

age[5] = JOHN_AGE

Rocket Academy

Shopee CODE LEAGUE 2020

Problems and Caveats

1. What if hash function is not uniform?

2. Even if it is uniform, if N gets sufficiently large, we still need to traverse a Linked List n/8 times, resulting in O(N) complexity.

Problems and Caveats

1. What if hash function is not uniform?
   a. Design of good hash functions
   b. Uniform Hashing Assumption: Any given element is equally likely to hash into any of the l slots, independent of where any other element has hashed to
   c. Enrichment: Design of good hash functions for different use case (UHA, speed etc)

2. Even if it is uniform, if N gets sufficiently large, we still need to traverse a LL n/8 times, resulting in O(N) complexity.
   a. Double the size of the table from 8 -> 16 to keep a constant load factor, l = N/size
   b. Assuming UHA, and a **load factor of 4**, how many times must a hash table of **size 8** double itself if we insert **200 unique key-value pairs**?
   c. How about **900 unique key-value pairs**
   d. Enrichment: Amortization cost of resizing

Rocket Academy

Shopee CODE LEAGUE 2020