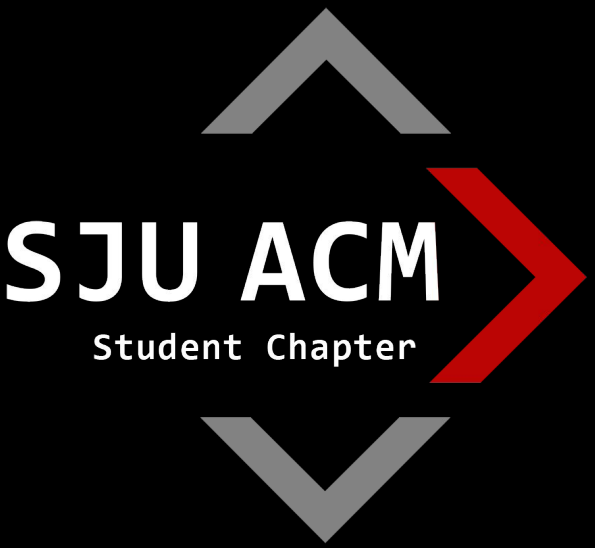
PYTHON TECHNICAL INTERVIEWS PART II

SJU ACM STUDENT CHAPTER



WHY STUDY FOR TECHNICAL INTERVIEWS?

WHAT LANGUAGE SHOULD YOU USE?

PYTHON VS JAVA FOR LOOP SYNTAX

PYTHON

```python
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
for x in range(len(fruits)):
    print(x)
for i, fruit in enumerate(fruits):
    print(i, fruit)
```

```
apple
banana
cherry
0
1
2
0 apple
1 banana
2 cherry
```

JAVA

```java
public class Main {
  public static void main(String[] args) {
    String[] fruits = new String[]{"apple", "banana", "cherry"};
    for (int i = 0; i < fruits.length; i++) {
        System.out.println(fruits[i]);
    }
    for (int i = 0; i < fruits.length; i++) {
        System.out.println(i);
    }
    for (int i = 0; i < fruits.length; i++) {
        System.out.println(i + " " + fruits[i]);
    }
  }
}
```

```
apple
banana
cherry
0
1
2
0 apple
1 banana
2 cherry
```

EFFICIENCY : O(N)

PYTHON VS JAVA HASHMAP/DICTIONARY SYNTAX

PYTHON

```
dictionary = { 1:"integer", 2.03:"Decimal", "Lion":"Animal"}
```

OR

```
hashmap = {}
hashmap["a"] = 1
hashmap[4] = 3.67
hashmap[3.07] = "value"
```

JAVA

```
import java.util.HashMap;

public class Main {
  public static void main(String[] args) {
    // Create a HashMap object called capitalCities
    HashMap<String, String> capitalCities = new HashMap<String, String>();

    // Add keys and values (Country, City)
    capitalCities.put("England", "London");
    capitalCities.put("Germany", "Berlin");
    capitalCities.put("Norway", "Oslo");
    capitalCities.put("USA", "Washington DC");
    System.out.println(capitalCities);
  }
}
```

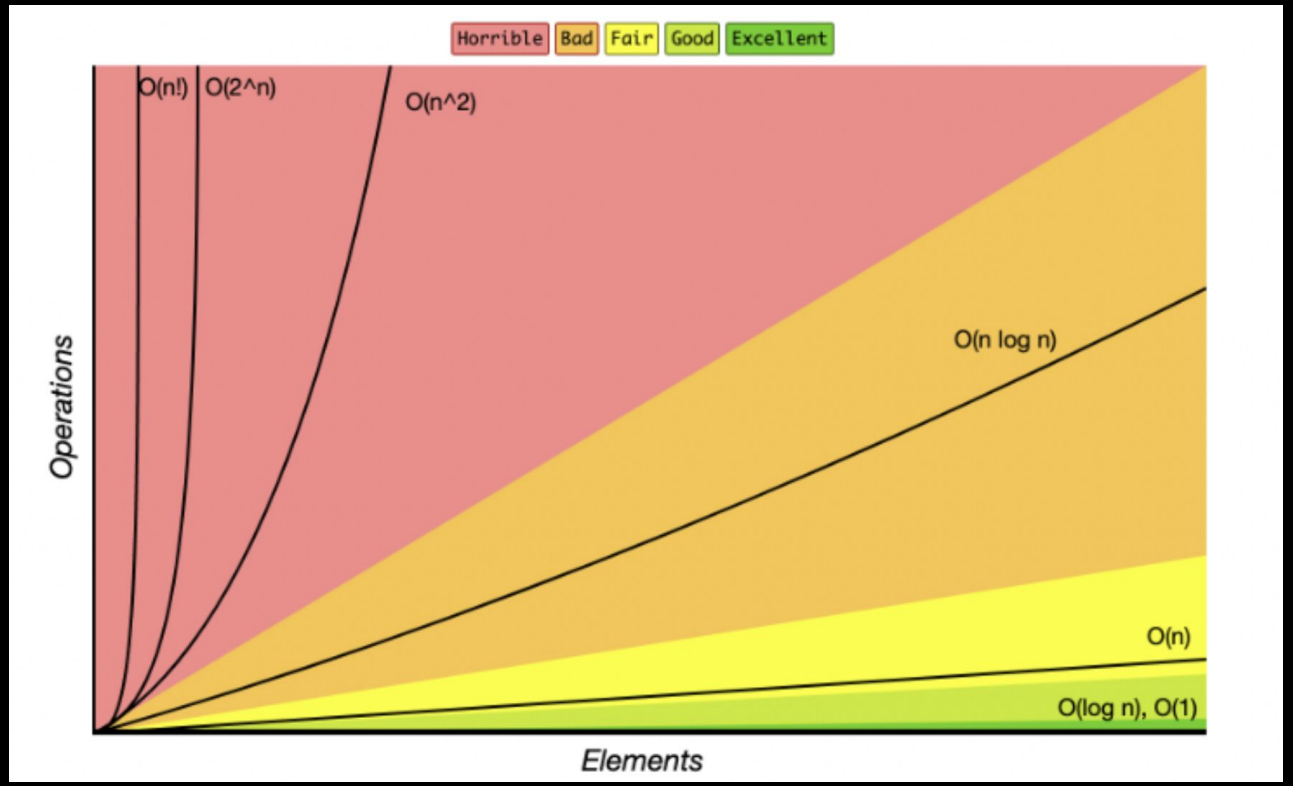
EFFICIENCY : O(1)

HOW DO YOU GET STARTED?

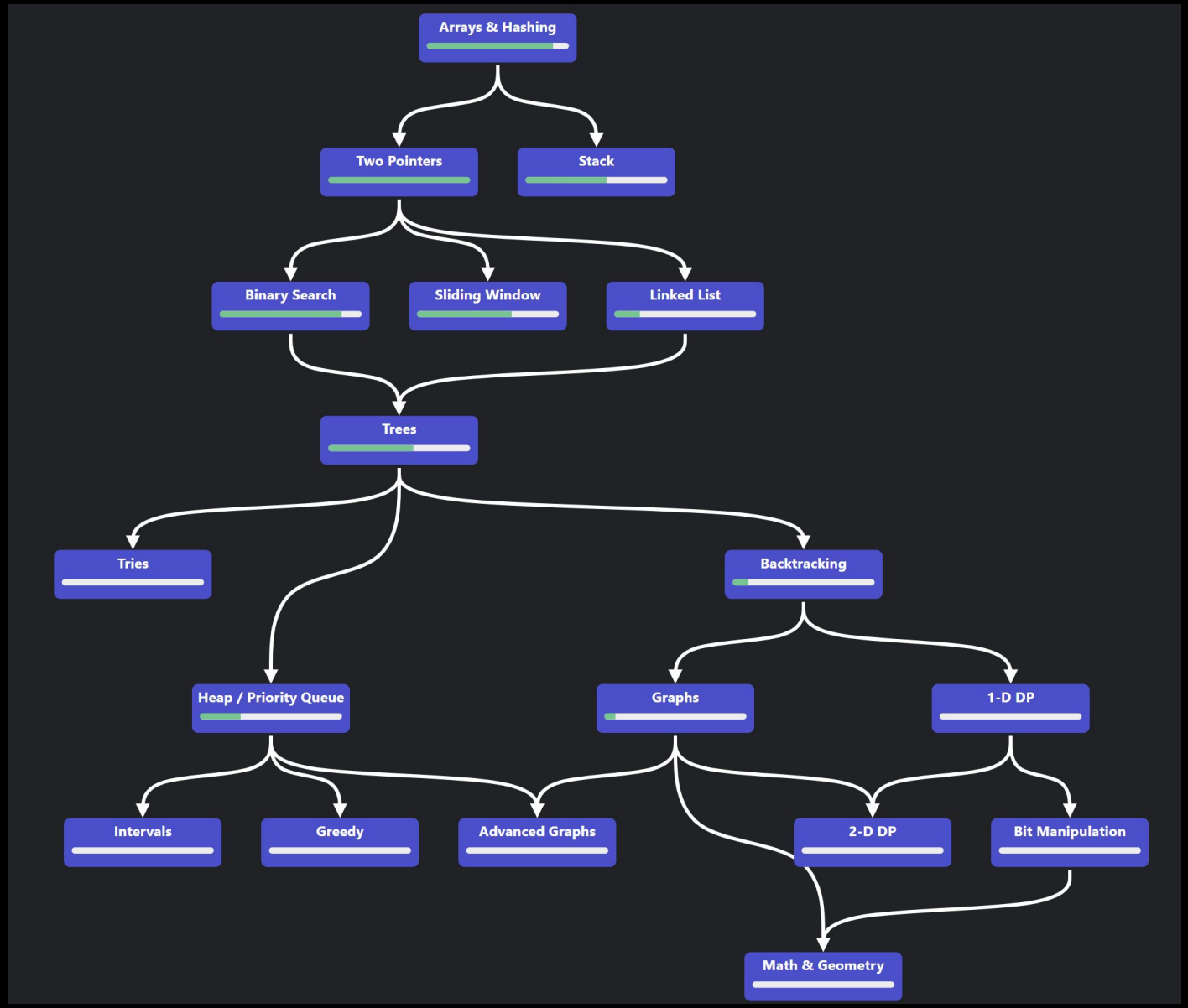
BIG O EFFICIENCY

IN BIG O, THERE ARE SIX MAJOR TYPES OF COMPLEXITIES (TIME AND SPACE):

- CONSTANT: O(1)
- LOGARITHMIC TIME: O(LOG N)
- LINEAR TIME: O(N)
- QUADRATIC TIME: O(N^2)
- EXPONENTIAL TIME: O(2^N)
- FACTORIAL TIME: O(N!)



NEETCODE.IO



PYTHON CHEAT SHEET

bit.ly/sjupython

VALUABLE PROBLEMS FOR INTERVIEWS

SIMPLE TWO POINTERS: 2 SUM SORTED:
HTTPS://LEETCODE.COM/PROBLEMS/TWO-SUM-II-INPUT-ARRAY-IS-SORTED/DESCRIPTION/

MEDIUM TWO POINTERS: CONTAINER WITH MOST WATER:
HTTPS://LEETCODE.COM/PROBLEMS/CONTAINER-WITH-MOST-WATER/DESCRIPTION/

SIMPLE SLIDING WINDOWS: BEST TIME TO BUY AND SELL STOCK:
HTTPS://LEETCODE.COM/PROBLEMS/BEST-TIME-TO-BUY-AND-SELL-STOCK/DESCRIPTION/

MEDIUM SLIDING WINDOWS: PERMUTATION IN STRING
HTTPS://LEETCODE.COM/PROBLEMS/PERMUTATION-IN-STRING/

THANK YOU!