DSA with Python - Cheat Sheet

Basic Python Syntax

```
# List
arr = [1, 2, 3]

# Dictionary
d = {"a": 1, "b": 2}

# Set
s = set([1, 2, 3])

# Tuple
t = (1, 2, 3)
```

Time Complexities

Operation	List	Set / Dict
Insert	O(1)	O(1)
Search	O(n)	O(1)
Delete	O(n)	O(1)
Sort	O(nlogn)	-

Loops & Comprehensions

```
# For Loop
for i in range(5): print(i)

# List Comprehension
squares = [x*x for x in range(5)]

# Dictionary Comprehension
d = {i: i*i for i in range(5)}
```

Common Data Structures

```
# List
arr.append(10)
arr.pop()
arr.sort()
arr.reverse()

# Set
s.add(4)
s.remove(2)
s.union({5})
s.intersection({3})
```

DSA with Python - Cheat Sheet

```
# Dict
d['key'] = 'value'
d.get('key', default)
d.keys(), d.values(), d.items()

# Heap
import heapq
heap = []
heapq.heappush(heap, 3)
heapq.heappop(heap)
```

Key Algorithms

```
# Binary Search
def binary_search(arr, target):
    l, r = 0, len(arr)-1
    while l <= r:
        m = (l + r) // 2
        if arr[m] == target:
            return m
        elif arr[m] < target:
            l = m + 1
        else:
            r = m - 1
    return -1

# Two Pointer
def has_pair_sum(arr, k):
    arr.sort()
    l, r = 0, len(arr)-1
    while l < r:
        if arr[l] + arr[r] == k:
            return True
        elif arr[l] + arr[r] < k:
            l += 1
        else:
            r -= 1
    return False
```

Sorting Algorithms

```
# Bubble Sort
def bubble_sort(arr):
    for i in range(len(arr)):
        for j in range(0, len(arr)-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

# In-built
sorted_arr = sorted(arr)
```

```
arr.sort()
```

Tree Traversals

```
# Inorder
def inorder(root):
    if root:
        inorder(root.left)
        print(root.val)
        inorder(root.right)
```

Linked List Basics

```
class Node:
    def __init__(self, val):
        self.val = val
        self.next = None

# Traversal
def print_list(head):
    while head:
        print(head.val, end=" -> ")
        head = head.next
```

Recursion

```
def factorial(n):
    if n == 0: return 1
    return n * factorial(n-1)
```

Useful Python Libraries

```
from collections import deque, Counter, defaultdict
import heapq
```