ChatGPT

Day 1 Notes: Google L3 Prep with Python

Folder: `Day1/`

`two_sum.py`

Problem: Two Sum – LeetCode 1

📚 Concepts:

- Hashmap (dictionary) for fast lookup
- Complement-based searching: `target - num`
- Return indices

Logic:

```
1. Create an empty dictionary `d`.
2. Loop through the list with index:
   - Calculate `complement = target - nums[i]`
   - If complement exists in `d`, return [d[complement], i]
   - Else add current num to dict with index as value
```

⏳ Time & Space:

- Time: `O(n)`
- Space: `O(n)`

`valid_palindrome.py`

Problem: Valid Palindrome – LeetCode 125

📚 Concepts:

- String cleaning: keep only alphanumeric
- Convert to lowercase
- Two-pointer method OR reversed comparison

Logic:

```
1. Filter the string: keep only letters/digits, convert to lowercase.
2. Compare cleaned string to its reverse.
3. If equal, return True.
```

⏳ **Time & Space:**

- Time: `O(n)`
- Space: `O(n)` (for cleaned string)

`longest_substring.py`

Problem: Longest Substring Without Repeating Characters – LeetCode 3

📚 **Concepts:**

- Sliding window technique
- Hashmap to track character index
- Dynamic window resizing

Logic:

```
1. Use a dictionary `seen` to store last index of each character.
2. Use two pointers: `start` and `i`.
3. If char is in seen and its index ≥ start, move start to seen[char] + 1.
4. Update max length.
5. Store/Update char in seen dict.
```

⏳ **Time & Space:**

- Time: `O(n)`
- Space: `O(n)`

Folder Summary:

```
Day1/
├── two_sum.py              # Hashmap + complement logic
├── valid_palindrome.py     # Clean string + reverse check
└── longest_substring.py    # Sliding window + dict
```

Next: [Day 2 Notes → Sets & Dictionary Grouping Problems]