COMPUTER SCIENCE CLUB

Intro to Competitive Programming (Pt. 5)

Edward Wang

What is a data structure?

A data structure is a way of representing data in a way that allows for certain specialized tasks

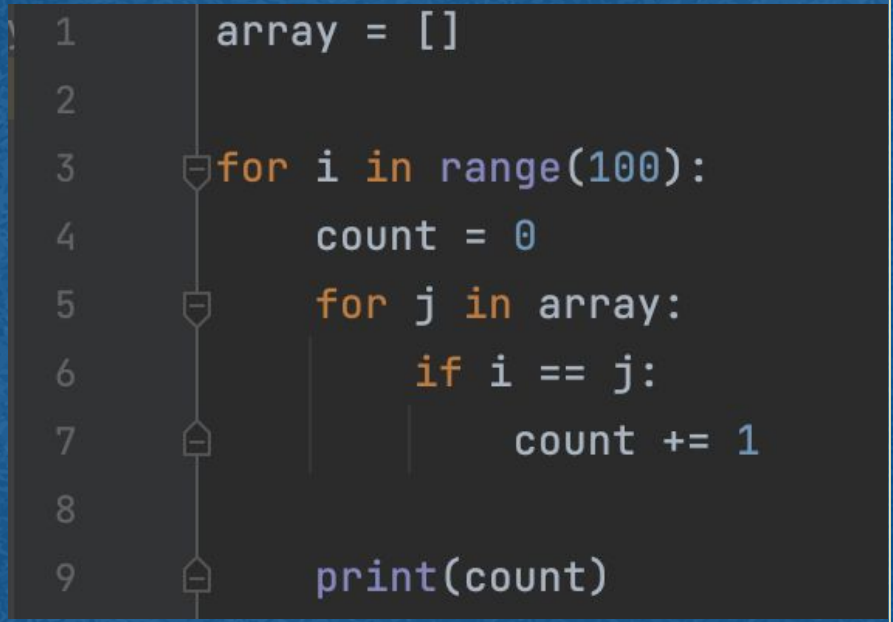
Examples:

- String
- List
- Dictionary
- Set

Problem:

Given an array of n integers, find how many times each value from 0-100 occurs.

Naive: Use .count() or for loops: O(n) per value



```
array = []

for i in range(100):
    count = 0
    for j in array:
        if i == j:
            count += 1

    print(count)
```

Frequency Array

Stores the amount of element n in the original at index n of the new array

Freq_array[5] evaluates to how many ‘5’s are in the original array

```
array = [5, 5, 111, 123, 124, 10, 1, 0, 111, 5]
# initialize the array
freq_array = [0] * max(array)

for i in array:
    # built the frequency array
    freq_array[i] += 1

# How many '5's are in the original array
print(freq_array[5])

# How many '111's are in the original array
print(freq_array[111])
```

Frequency Array

Initializing array:
Create an array with the length max(array), filled with 0's
Iterate through original array, add one to the index corresponding to the value

```
array = [5, 5, 111, 123, 124, 10, 1, 0, 111, 5]
# initialize the array
freq_array = [0] * max(array)

for i in array:
    # built the frequency array
    freq_array[i] += 1

# How many '5's are in the original array
print(freq_array[5])

# How many '111's are in the original array
print(freq_array[111])
```

Frequency Array Example

i	0	1	2	3	4	5	6	7	8
Array[i]	6	1	2	4	1	8	2	4	1
Freq[i]	0	3	2	0	2	0	1	0	1

Index 1 of the frequency array shows how many 1's are in the original array

Frequency Array

Initializing frequency array: max(array) elements, O(max(array)) or O(n) with dictionary

```
freq_array = [0] * max(array)
```

Query frequency array: array access, O(1) time

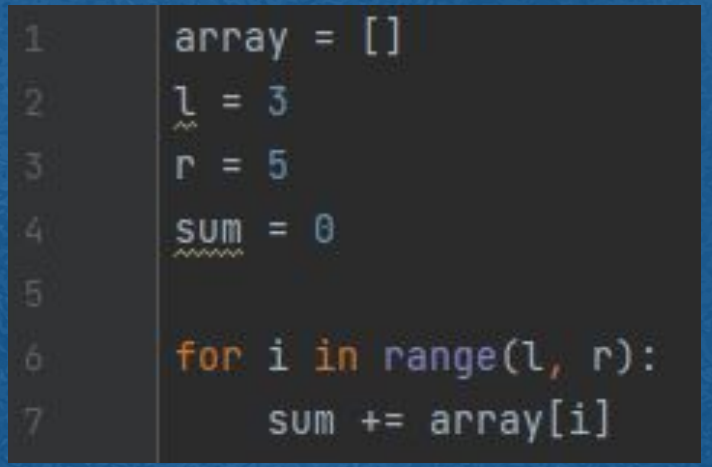
```
# How many '5's are in the original array
print(freq_array[5])

# How many '111's are in the original array
print(freq_array[111])
```

Problem:

Given an array of n integers, find the sum from index l to index r

Naive: Use sum() or for loop iteration



```
array = []
l = 3
r = 5
sum = 0

for i in range(l, r):
    sum += array[i]
```

Prefix Sum Array

Stores the sum from index 0 to index i at every index i

psa[5] evaluates to the sum from elements 0 to 5

```
array = [1, 5, 2, 3, 5]
psa = []

for i in array:
    if psa:
        psa.append(psa[-1] + i)
    else:
        psa.append(i)

print(psa)
```

```
[1, 6, 8, 11, 16]
```

Prefix Sum Array

To find value at index i, find the difference between index i and index i-1

To find sum in a range (inclusive), psa[r] - psa[l]

```python
array = [1, 5, 2, 3, 5]
psa = []

for i in array:
    if psa:
        psa.append(psa[-1] + i)
    else:
        psa.append(i)

print(psa)
```

```
[1, 6, 8, 11, 16]
```

Prefix Sum Array

Initializing PSA: O(n) time, iterating through original array

Query sum in PSA: O(1) time

Practice Problems

Beginner (PSA):
https://dmoj.ca/problem/dmopc14c2p4
Intermediate (PSA):
https://dmoj.ca/problem/sac21ccp4
Advanced (frequency array):
https://dmoj.ca/problem/ccc21s2
https://dmoj.ca/problem/ccc17s3