**Justin\_Design\_Queue\_0346.**  **Moving Average from Data Stream**

**Concept:**

len(self.data) < self.size 就直接加入，再算出平均

len(self.data) = self.size 就先刪除第一個再加入，最後算出平均

**Code:**

class MovingAverage:

def \_\_init\_\_(self, size: int):

"""

Initialize your data structure here.

"""

self.data = []

self.size = size

def next(self, val: int) -> float:

result = 0

if len(self.data) < self.size:

self.data.append(val)

for i in range(len(self.data)):

result += self.data[i]

return result/len(self.data)

elif len(self.data) == self.size:

del self.data[0]

self.data.append(val)

for i in range(len(self.data)):

result += self.data[i]

return result/len(self.data)

# Your MovingAverage object will be instantiated and called as such:

# obj = MovingAverage(size)

# param\_1 = obj.next(val)