
DSC 40B - Homework 05

Due: Wednesday, February 12

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 p.m.

This homework is shorter – just one question – due to the midterm exam

Programming Problem 1.

Recall that a *mode* of a collection is an element which occurs with the greatest frequency. For example, 4 is a mode of the collection 4, 5, 8, 3, 4, 2, 4, 5, 5, −2. 5 is also a mode, since it occurs just as frequently as 4.

In a file named `mode.py`, create a function named `mode(numbers)` which takes in an array of numbers and returns a mode. To receive full credit, your function should have the best possible average case time complexity. If there are multiple modes, your function need only return one of them. You should *not* assume that the numbers are integers or that they are positive.

Note: it's up to you to determine the best possible time complexity and convince yourself that your code is efficient! Think back to when we talked about theoretical lower bounds. Can you come up with one for this problem? This is a situation you'll find yourself in often when writing code: there's usually no resource that tells you what the best time complexity is for a given problem, so you'll have to convince yourself that your solution is optimal.

Solution:

```
def mode(numbers):
    # keep track of counts in a dictionary
    counts = {}

    for number in numbers:
        if number not in counts:
            counts[number] = 1
        else:
            counts[number] += 1

    # this returns the dictionary key that has the largest value
    return max(counts, key=lambda key: counts[key])
```