Brendan Sherman

Prof. Alvarez

CSCI 229

January 24, 2022

Homework 1 Master Writeup

1. In each case below, write a single Python expression that has the stated value.

   a. The smallest among the squares of the elements of the list L.

   ```
   min([L[n]**2 for n in range(len(L))])
   ```

   Explanation: The outer function, "min", returns the smallest value within the list which is passed to it. Using list comprehension, the list passed to "min" here contains the squares of each element within L by iterating through it and squaring each value. Therefore, the entire expression represents the smallest among the squares of elements in list L.

   b. The second-from-last element (third highest index position) of the numpy array a. The length of a is not known in advance (but it's at least 3).

   ```
   a[-3]
   ```

   Explanation: This expression indexes backwards into the numpy array a, with negative 3 being equivalent to the third highest index position. Regardless of the "shape" of the array, indexing like this will access the second-from-last element of the base list of a (whether that element is another list or a value).

   c. The sum $\sum_{n=-50}^{10^4} n^3$

   ```
   sum([n**3 for n in range(-50, (10**4) + 1)])
   ```

   Explanation: The outer function, "sum", returns the sum of all values within the list which is passed to it. Therefore, here sum is passed a list containing the cubes of all values between -50 and 10**4 (inclusive as per summation notation, which is why the "+1" is needed). Therefore, the final expression is representative of the given summation, evaluating to the correct value of 2500500023374375.

   d. A list of all non-negative integers n < 100 such that $n^4 > 500n$.

   ```
   [n for n in range(100) if (n**4) > (500*n)]
   ```

   Explanation: Using list comprehension, this expression creates a list by iterating between 0 and 99 (inclusive), adding them to the list if the value *n**4* (n^4) is greater than *500*n* (500n). Therefore, the value of this expression is all non-negative integers less than 100 where $n^4 > 500n$.

e. The number of entries in the dictionary D that do not have the value 'red'.

```
len([v for v in D.values() if v != "red"])
```

Explanation: Using list comprehension, this expression creates a list by iterating through all values in dictionary D (accessed with "D.values()") and adding them if the value is not equal to "red". This list is then passed into len(), providing the length of this list, equivalent to the number of entries in D that do not have "red" as their value.

2. A certain real-valued function of a real variable, f(x), has the value f(π) = 1 and the derivative f′(x) = 1 −sin(x). Find the numerical value of f(0). Explain.

f(0) = 3 – π ≈ -0.14159265359

Explanation: Because we are given the derivative and a specific value of f, we can find the exact function by using antiderivatives and then solving for C. This allows us to then calculate f(0).

f(x) = $\int 1 - \sin(x)\, dx = x + \cos(x) + C$ (where C is some constant)

we are given f(π) = 1, so $\pi + \cos(\pi) + C = 1$

Solving for the unknown, we get C = $2 - \pi$

Therefore, $f(x) = x + \cos(x) + 2 - \pi$

Finally, we plug 0 into f(x) to get $f(0) = 0 + \cos(0) + 2 - \pi = 3 - \pi$

3. In each case below, write clear, simple Python code that will help you to determine the stated quantity.

a. The largest positive integer, m, for which $10^m < 1000m^6$

Answer: 8

Source Code:

```
def problem3a():

i = 1

while((10**i) < (1000 * (i**6))):

    i += 1

return (i-1)

def main():

    print(problem3a())
```

Explanation: The code above prints "8", indicating that 8 is the largest positive integer m such that $10^m < 1000m^6$. This solution works by iterating upwards through each positive integer (using *i* as the iterator) with a while loop. The condition for the loop to continue is the Python equivalent of the expression $10^i < 1000i^6$, meaning that it will continue iterating until said expression evaluates to false, indicating that the largest possible integer satisfying the expression is $i - 1$ (which was the greatest value to satisfy the condition). This has to be the greatest possible value because the function 10^m grows at a rate much faster than $1000m^6$, and therefore once the functions intersect, 10^m will always be greater than $1000m^6$ (for m>0). Therefore, after the loop executes the function returns $i - 1$, which is always equal to 8.

b. The largest positive integer, m, for which $1000 - 100m + 5m^2 - \frac{1}{15}m^3 \geq 500$.

<u>Answer:</u> 45

<u>Source Code:</u>

```
def problem3b():

i = 37

while((1000 - 100*i + (5 * (i**2)) - ((1/15)*(i**3))) >= 500):

      i += 1

return i-1

def main():

      print(problem3b())
```

<u>Explanation:</u> The code above prints “45”, indicating that 45 is the largest positive integer m such that $1000 - 100m + 5m^2 - \frac{1}{15}m^3 \geq 500$. Because $1000 - 100m + 5m^2 - \frac{1}{15}m^3$, which I will call f(m), crosses the line y=500 several times, we cannot start iterating at 1 because there may be a point where f(m)< 500 but then crosses y=500 again. Therefore, using sign analysis on the derivative, $(-100 + 10m - \frac{3}{15}m^2)$, I calculate that when m>36.18 (a zero of the derivative), f’(m) is always negative, and therefore the function f(m) is always decreasing. Furthermore, when m=36.18, f(m) > 500. Therefore, if we start iterating at i=37, we can be confident that eventually f(i) will intersect y=500, and that this is the largest possible value where the function is >= 500. Similar to part a, then, we iterate through integers with a loop until the python equivalent of $1000 - 100i + 5i^2 - \frac{1}{15}i^3 \geq 500$ evaluates to false, indicating that i-1, which always evaluates to 45, is the largest possible integer satisfying the expression.