HW1

Seth Jaksik
ID: 1001541359

August 28, 2020

Task 1

The time complexity of the factorial function is $\boxed{\Theta(n)}$.

Task 2

The recursive implementation of the factorial function of Task 1 is:

```
def factorial(n):
    if(n == 1 or n == 0):
        return 1
    return n * factorial(n-1)
```

Task 3

The time complexity of the foo function is $\boxed{\Theta(n^2)}$

Task 4

The result of matrix multiplication A*B is: $\boxed{\begin{bmatrix} ae + bf \\ ce + df \end{bmatrix}}$

Task 5

For the function: $f(x) = 3x^2 + 5x - 7$

Part A

The first derivative $f`(x) = \boxed{6x + 5}$

Part B

$f'(5) = 6(5) + 5 = \boxed{35}$

Part C

The second derivative $f''(x) = \boxed{6}$

Part D

$f''(5) = \boxed{6}$

Task 6

For $P(A) = 0.3$ *and* $P(B) = 0.6$:

- $P(AandB) = P(A) * P(B) = (0.3)(0.6) = \boxed{0.18}$

- $P(AorB) = P(A) + P(B) = (0.3) + (0.6) = \boxed{0.9}$

- $P(not(A)) = 1 - P(A) = 1 - (0.3) = \boxed{0.7}$

- $P(A|B) = \frac{P(AandB)}{P(B)} = \frac{(0.18)}{(0.6)} = \boxed{0.3}$

1   Task 7

Definition of the file_stats(pathname) function:

```
import math

def file_stats(pathname):
    avg = stdev = 0
    file = open(pathname, "r")
    nums = []
    for i in file.read().split('\n'):
        nums.append(float(i))

    # calc avg
    for i in range(len(nums)):
        avg += float(nums[i])
    avg /= len(nums)

    # calc stdev
    for i in range(len(nums)):
        stdev += (nums[i]-avg)**2
    stdev = math.sqrt(stdev / (len(nums) - 1))
    return avg, stdev
```

Task 8

For the table:

- $P(price < \$75) = \frac{num < \$75}{total} = \frac{255}{400} = \boxed{0.6375}$

- $P(price < \$75 \mid color = green) = \frac{P(price < \$75 \ and \ color = green)}{P(color = green)} = \frac{65/400}{95/400} = \boxed{0.6842}$

- $P(price < \$75, \ color = green) = \frac{num < \$75 \ and \ green}{total} = \frac{65}{400} = \boxed{0.1625}$