CSCI 2824 – Spring 2020 **Homework 6**

This assignment is due on Friday, February 28 to Gradescope by 11:59pm. You are expected to write up your solutions neatly and **use the coverpage**. Remember that you are encouraged to discuss problems with your classmates, but you must work and write your solutions on your own.

Important: On the **official CSCI 2824 cover page** of your assignment clearly write your full name, the lecture section you belong to (001 or 002), and your student ID number. You may **neatly** type your solutions for +2 extra credit on the assignment. You will lose *all* 5 style/neatness points if you fail to use the official cover page.

(1) In an in-class example, we observed anecdotally that for non-empty sets P, Q, and R, the following holds:

$$(P \cap Q) \times R = (P \times R) \cap (Q \times R)$$

(a) Prove that $(P \cap Q) \times R = (P \times R) \cap (Q \times R)$ holds by showing that each side of this equation is a subset of the other side of the equation.

(b) Prove that that $(P \cap Q) \times R = (P \times R) \cap (Q \times R)$ holds using **set builder** notation and using set identities.

Solution:

(a) Let $Z \in (P \cap Q) \times R$
$z = (x, y),\ x \in P \cap Q$ and $y \in R$
$(x \in P$ and $x \in Q)$ and $y \in R$
$(x \in P$ and $y \in R)$ and $(x \in Q$ and $y \in R)$
$(x, y) \in P \times R$ and $(x, y) \in Q \times R$
$(x, y) \in (P \times R) \cap (Q \times R)$
$z \in (P \times R) \cap (Q \times R)$
$\therefore (P \cap Q) \times R \subseteq (P \times R) \cap (Q \times R)$

Now, $z \in (P \times R) \cap (Q \times R)$
$z \in P \times R$ and $z \in Q \times R$
$(x, y) \in P \times R$ and $(x, y) \in Q \times R$
As before, $(x \in P$ and $x \in Q)$ and $y \in R$
$x \in P \cap Q$ and $y \in R$
$(x, y) \in (P \cap Q) \times R$
$z \in (P \cap Q) \times R$
$\therefore (P \times R) \cap (Q \times R) \subseteq (P \cap Q) \times R$

$\therefore (P \cap Q) \times R = (P \times R) \cap (Q \times R)$

(b) $(P \cap Q) \times R$
$= \{(x, y) : x \in P \cap Q,\ y \in R\}$
$= \{(x, y) : x \in P \text{and} x \in Q,\ y \in R\}$
$= \{(x, y) : x \in P,\ y \in R\}$ and $\{(x, y) : x \in Q,\ y \in R\}$
$= \{(x, y) : x \in P,\ y \in R\} \cap \{(x, y) : x \in Q,\ y \in R\}$
$= (P \times R) \cap (Q \times R)$

(2) Complete the following recurrence exercises:

(a) Find a closed form of the recurrence given by $a_n = 7 \cdot a_{n-1} - 5$; $a_0 = 4$

(b) Find a closed form of the recurrence given by $a_n = (n + 1)^2 \cdot a_{n-1}$; $a_0 = 1$

(c) Consider the recurrence $a_n = a_{n-1} + 2a_{n-2} + 2n - 9$.

(i) Show that this recurrence is solved by $a_n = 2 - n$.

(ii) Show that this recurrence is solved by $a_n = 2 - n + b \cdot 2^n$ for any real b.

Solution:

(a) $a_n - 7a_{n-1} = -5$
Express $a_n - 7a_{n-1} = -5$ for all $n = n, (n-1), \ldots, 1$
$a_n - 7a_{n-1} = -5$
$7a_{n-1} - 7^2 a_{n-2} = -5 \cdot 7$
$7^2 a_{n-2} - 7^3 a_{n-3} = -5 \cdot 7^2$
...
...
$7^{n-1} a_1 - 7^n a_0 = -5 \cdot 7^{n-1}$
$a_n - 7^n a_0 = -5 - 5 \cdot 7 - 5 \cdot 7^2 - \ldots - 5 \cdot 7^{n-1}$
$a_0 = 4$
$\therefore a_n - 4 \cdot 7^n = -5(1 + 7 + 7^2 + \ldots + 7^{n-1})$
$a_n - 4 \cdot 7^n = -5(\frac{1(7^n-1)}{7-1})$
$a_n = 4 \cdot 7^n - \frac{5(7^n-1)}{6}$
$a_n = \frac{24 \cdot 7^n - 5 \cdot 7^n + 5}{6}$
$a_n = \frac{19 \cdot 7^n + 5}{6}$

(b) $a_n = (n+1)^2 \cdot n^2 \cdot a_{n-2}$
$= (n+1)^2 \cdot n^2 \cdot (n-1)^2 \cdot a_{n-3}$
$= (n+1)^2 \cdot n^2 \cdot (n-1)^2 \ldots 2^2 \cdot a_0$
$a_n = (n+1)^2$

(c) (i) $a_n = 2 - n = 2 - (n-1) + 2(2 - (n-2)) + 2n - 9$
$2 - n = 3 - n + 8 - 2n + 2n - 9$
$2 - n = 2 - n$

(ii) $a_n = 2 - n + b \cdot 2^n = 2 - (n-1) + b \cdot 2^{n-1} + 2(2 - (n-2) + b \cdot 2^{n-2}) + 2n - 9$
$= 2 - n + b \cdot n^{n-1} + 2b \cdot 2^{n-2}$
$= 2 - n + 2b \cdot 2^{n-1}$
$= 2 - n + b \cdot 2^n$

(3) Consider the function $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ given by $f(n, m) = \frac{n^3}{|m|+1}$

(a) Write in logical predicate notation - using quantifiers for n, m, etc. as appropriate - the logical equivalent of the function f being onto.

(b) Determine whether or not f is one-to-one.

(c) Determine whether or not f is onto.

Solution:

(a) f is onto iff:
$\forall y \in \mathbb{Z}\ \exists (n, m) \in \mathbb{Z} \times \mathbb{Z}$ such that $f(n, m) = y$

(b) $f(0, 1) = 0$ and $f(0, 2) = 0$
$f(0, 1) = f(0, 2)$ but $(0, 1) \neq (0, 2)$
f **IS NOT** one-to-one

(c) define $y \in \mathbb{Z}$ then observe:
$f(y, y) = \frac{y^3}{|y^2-1|+1} = y$
$\therefore \forall y \in \mathbb{Z}\ \exists (n, m) \in \mathbb{Z} \times \mathbb{Z}$ such that $f(n, m) = y$
$\therefore f$ **IS** onto

(4) Define the set C = the set of all Spring 2020 CSCI2824 students.

(a) Define in words a function $f : C \to \mathbb{Z}$. Is the function function one-to-one and/or onto? Be sure that f is actually a function, and feel free to be creative!

(b) Again define the set C = the set of all Spring 2020 CSCI2824 students. Define in words another function $g : C \to \mathbb{Z}$, but ensure that g is definitely one-to-one.

Solution:

(a) def C = the set of all CSCI2824 students as stated above
def $f : C \to \mathbb{Z}$ as student $\to$ their height
This is a function, however it **WILL NOT BE ONTO** as the range is finite and the codomain is infinite.
Given the fact two or more people in the class are most likely the same height, this function **WILL NOT BE ONE-TO-ONE**.

(b) def C = the set of all CSCI2824 students as stated above
def $f : C \to \mathbb{Z}$ as student $\to$ their Social Security Number
This is a function, and each student is mapped to their individual and unique SSN, meaning this function **WILL BE ONE-TO-ONE**

(5) The given code attempts to answer the popular debate: "which Jedi wins in a duel?"
The code:

```python
def WhoWins(Jedi, LSColors, Heights):
    if len(Jedi)!= len(LSColors) or len(Jedi)!= len(Heights) or len(Heights)!= len(LSColors):
        print('Error! Vector Length mismatch')
        return
    for i in range(len(Jedi)):
        for j in range(len(Jedi)):
            #Sam Jackson always wins:
            if LSColors[i]=='Purple':
                winner=Jedi[i]
            elif LSColors[j]=='Purple':
                winner=Jedi[j]
            #Does one have the high ground?
            elif Heights[i]/Heights[j]>1:
                winner=Jedi[i]
            elif Heights[i]/Heights[j]<1:
                winner=Jedi[j]
            else:
                winner='a tie'
            print('For ', Jedi[i], ' against ', Jedi [j], ", the winner is ", winner)
```

i) Makes sure the inputs - Jedi names, lightsaber colors, and current topographic information - are the same sizes.

ii) Pairs up Jedi within a couple of for loops

iii) Checks if Jedi #1 is Mace or if Jedi #2 is Mace. He always wins in these debates.

iv) Checks which Jedi is currently at the highest elevation. After all, the high ground wins!

You, aspiring Jedi enthusiast, must answer:

(a) What is the **algorithmic complexity** of this code? In other words, exactly how comparisons are checked if n Jedi are input correctly? You may assume that *each* part of the **if, elif, else** statement does in fact generate a comparison regardless of input, as this is a "worst-case" analysis.

(b) What are some redundancies of this code? Could it be done in less comparisons?

(c) (Not for points): In your opinion, what should actually have been used to determine the winners?

```
Jedi=['Anakin', 'Obi-Wan']
LSColors=['Blue', 'Blue']
Heights=[5.5,5.8]
WhoWins(Jedi, LSColors, Heights)
```

```
For  Anakin  against  Anakin , the winner is  a tie
For  Anakin  against  Obi-Wan , the winner is  Obi-Wan
For  Obi-Wan  against  Anakin , the winner is  Obi-Wan
For  Obi-Wan  against  Obi-Wan , the winner is  a tie
```

Solution:

(a) Assuming the 3 lists are inputted correctly, the code would perform $5n^2$ comparisions in total, as there are 5 comparisons with in a nested for loop iterating n times.

(b) Defining the 'winner' variable as 'a tie' at the beginning saves one comparison, then you simply reassign it if either of the Jedi do win. This would take the comparisons in each loop down to 4. The code would end up looking something like:

```python
def WhoWins(Jedi, LSColors, Heights):
    if len(Jedi) != len(LSColors) or len(Jedi) != len(Heights) or len(Heights) != len(LSColors):
        print("Error! Vector Length mismatch")
        return

    for i in range(len(Jedi)):
        for j in range(len(Jedi)):
            winner = 'a tie'

            if LSColors[i] == 'Purple' or Heights[i] > Heights[j]:
                winner = Jedi[i]
            if LSColors[j] == 'Purple' or Heights[j] > Heights[i]:
                winner = Jedi[j]

            # only 4 comparisons in each loop
            print('For', Jedi[i], ' against', Jedi[j], ', the winner is', winner)
```

(c) I agree with taking lightsaber color into account, but everyone knows that the coolest lightsaber color is yellow, therefore that should be the winning color. Instead of height, the rank of the Jedi in question should determine who would win in every other case, with the highest rank Jedi winning and a tie resulting if two Jedi are the same rank.