Problem Set 0

Name: Moaaz Elsayed

Problem 0-1.

$A = \{1, 6, 12, 13, 9\}$
$B = \{3, 6, 12, 15\}$

(a)

$A \cap B = \{6, 12\}$

(b)

$A \cup B = \{1, 6, 9, 12, 13, 9, 15\}$
$|A \cap B| = 7$

(c)

$A - B = \{1, 12, 9\}$
$|A - B| = 3$

Problem 0-2.

$X \in \{0, 1, 2, 3\}$
$Y \in \{1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 30, 36\}$

(a)

$$
\begin{aligned}
E|X| &= \sum_{i=0}^{3} x_i P(x_i) \\
p &= \frac{1}{2} \\
p(x_i) &= \binom{n}{x} p^{x_i} (1-p)^{n-x_i} \\
\therefore E|X| &= \frac{3}{2}
\end{aligned}
$$

(b)

$$
\begin{aligned}
E|Y| &= \sum_{i=0}^{18} y_i P(y_i) \\
p(Y=1) &= \frac{1}{36} \\
p(Y=2) &= \frac{2}{36} \\
p(Y=3) &= \frac{2}{36} \\
&\vdots \\
E|Y| &= \frac{49}{4}
\end{aligned}
$$

(c)

$$
E|X+Y| = E|X| + E|Y| = \frac{55}{4}
$$

Problem 0-3.

$A = 100$
$B = 18$

(a) True
(b) False
(c) False

Problem 0-4.

$$\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 \tag{1}$$

we can prove it **by induction**:
P(n) will be equation (1).
Base case: $n = 0$ is true, because both sides equals 0.
Inductive case: assume that P(n) is true,where n is any nonnegative integer.

$$\begin{aligned}\sum_{i=1}^{n+1} i^3 &= \left(\frac{n(n+1)}{2}\right)^2 + (n+1)^3 \\ &= \frac{n^4 + 6n^3 + 13n^2 + 12n + 1}{4} \\ &= \frac{(n+1)^2(n+2)^2}{4} \\ &= \left(\frac{(n+1)(n+2)}{2}\right)^2.\end{aligned}$$

which proves P(n + 1).
So it follows by induction that P(n) is true for all nonnegative n.

$\square$

Problem 0-5.

$\because$ G is connected

$\therefore$ there is a path between every two vertices, evey edge is connected to at least one edge. (2)

$\because |E| = |V| - 1$

$\therefore$ evey vertex is connected to at most one edge.. (3)

$\because$ there must be at least one edge with a degree of two in order to make a cycle.

$\therefore$ 2 and 3 the graph is acyclic.

Problem 0-6. Submit your implementation to `alg.mit.edu`.

```
def count_long_subarray(A):
    '''
    Input:  A     | Python Tuple of positive integers
    Output: count | number of longest increasing subarrays of A
    '''
    count = 1
    curr_count = 1
    max = 1
    for i in range(1, len(A)):
        # print(A[i - 1])
        # print(A[i])
        if A[i] > A[i - 1]:
            curr_count += 1
            # if curr_count > max:
        else:
            curr_count = 1
        if curr_count == max:
            count += 1
        elif curr_count > max:
            max = curr_count
            count = 1
    return count
```