Problem Set 0

All parts are due on September 8, 2019 at 6PM. Please write your solutions in the LaTeX and Python templates provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on Gradescope, and any code should be submitted for automated checking on `alg.mit.edu`.

This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. You **may** ask for logistical help concerning LaTeX formatting and/or code submission.

Problem 0-1. Let $A = \{2^i \mid i \in \mathbb{N} \text{ and } 0 \leq i < 5\}$ and $B = \{2i - 1 \mid i \in \{0, 1, 2, 3\}\}$.

Evaluate: **(a)** $A \cap B$ **(b)** $|A \cup B|$ **(c)** $|A - B|$

Solution: (a) $\{1\}$, (b) 8, (c) 4

Common Mistakes:

- Providing the sets for (b) and (c) instead of their cardinality
- Including 2^5 in A

Problem 0-2. Let X be the random variable representing the number of blue balls chosen, after choosing two balls uniformly at random without replacement from a bag containing exactly three blue balls and two red balls. Let Y be the random variable representing the number of heads seen after flipping a fair coin twice.

Evaluate: **(a)** $\mathrm{E}[X]$ **(b)** $\mathrm{E}[Y]$ **(c)** $\mathrm{E}[X + Y]$

Solution: (a) $6/5 = 1.2$, (b) 1, (c) $11/5 = 2.2$

Common Mistakes:

- Not knowing what an expected value is
- Thinking that $\mathrm{E}[X + Y] = \mathrm{E}[X]\,\mathrm{E}[Y]$ instead of $\mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y]$.

Problem 0-3. Let $A = 600 + 6$ and $B = 60 \times (4 + 2)$. Are these statements True or False?

Evaluate: **(a)** $A \equiv B \bmod 2$ **(b)** $A \equiv B \bmod 3$ **(c)** $A \equiv B \bmod 4$

Solution: (a) True, (b) True, (c) False

Problem 0-4. Prove **by induction** that $\sum_{i=0}^{n} a^i = \frac{1-a^{n+1}}{1-a}$ when $a \neq 1$, for any integer $n \geq 0$.

Solution: Induct on n. Base Case: for $n = 0$, $a^0 = 1 = \frac{1-a^1}{1-a}$ as desired. Now assume by induction that $\sum_{i=0}^{n} a^i = \frac{1-a^{n+1}}{1-a}$ is true for $n = k$. We prove the statement is true for $n = k + 1$:

$$\sum_{i=0}^{k+1} a^i = a^{k+1} + \sum_{i=0}^{k} a^i = a^{k+1} + \frac{1-a^{k+1}}{1-a} = \frac{(1-a)a^{k+1} + (1-a^{k+1})}{1-a} = \frac{1-a^{k+2}}{1-a}, \text{ as desired.} \square$$

Problem 0-5. Prove **by induction** that the vertices of a tree (a connected acyclic undirected graph) can each be colored either red or blue such that no edge connects two vertices of the same color. You may use the fact that every tree having more than one vertex contains a vertex with degree one.

Solution: Induct on the number of vertices k. Base Case: a tree containing one vertex has no edges, so can be colored either color, as desired. Now assume for induction the claim is true for any tree on k vertices, and consider any tree T containing exactly $k+1$ vertices. By the fact above, the tree contains a vertex v connected to exactly one edge to another vertex u. Removing v and the edge connecting v to u yields a tree T' on k vertices. By the inductive hypothesis, the vertices of T' can be colored red or blue such that no edge connects two vertices of the same color. Whatever color u is colored in this coloring, coloring v the other color provides a coloring of T such that no edge connects two vertices of the same color, as desired. $\square$

Common Mistakes:

- Arguing base case $|V| = 2$ and not $|V| = 1$
- Arguing for induction that, given a graph with k vertices, adding a leaf will maintain the claim. You need to show the claim holds for any graph with $k+1$ vertices, not that the claim works for any graph constructed from a graph on k vertices. This was a very common mistake.

Problem 0-6. Write a Python function `min_mod_tuple(A, k)` which accepts two arguments, Python List `A` $= [a_0, a_1, \ldots, a_{n-1}]$ containing n positive integers and positive integer `k`, and returns a Python Tuple (i, j) of two array indices with $0 \leq i < j < n$ that minimizes $(a_i \times a_j) \bmod k$. You can download a code template containing some test cases from the website. Submit your code online at `alg.mit.edu`.

Solution:

```python
def min_mod_tuple(A, k):
    n = len(A)
    i, j, small = 0, 1, (A[0] * A[1]) % k
    for j_ in range(1, n):
        for i_ in range(j_):
            prospect = (A[i_] * A[j_]) % k
            if prospect < small:
                i, j, small = i_, j_, prospect
    return (i, j)
```