Problem Set 1 Solutions

Problem 1. *Alice has a pretty penny. Unfortunately, it might not be a fair penny: it might, when flipped, land heads with some probability $p \neq 0.5$. Alice wants to generate a uniform random bit b: the bit should be 1 with probability 0.5 and zero with probability 0.5. Describe a strategy Alice can use to achieve the result she wants using her possibly-biased coin.*

We assume that coin flips are all independent of one another and that $0 < p < 1$ (absent these assumptions, a solution is impossible). Alice can then flip the coin until she sees the first time that it goes from an outcome of heads to an outcome of tails or, conversely, it goes from an outcome of tails to an outcome of heads. (With probability 1, one of these events will eventually occur, since $0 < p < 1$.) In the first case she outputs "1" and in the second case she outputs "0". By independence, both occur with equal probability.

Problem 2. *Alice and Bob have an infinite pile of pennies. They take turns placing their pennies on a perfectly round table, beginning with Alice. A penny may be placed anywhere on the table so long as all of the penny fits fully on top of the table and no part of the penny is on top of any other penny. Pennies must be placed flat on their heads or tails side. A party loses if he has nowhere to put his penny. Show that Alice can always win. (You might need some natural assumption for this to be true. If so, state it.)*

We must assume that the table's diameter exceeds the penny's diameter; otherwise, contrary to the problem statement, Alice loses. Here's Alice's strategy. She places her penny exactly in the center of the table. Then, whenever Bob places a penny on the table, Alice places one at the "opposite" location—meaning that the center of the penny is on the line connecting the center of Bob's newly placed penny and the center of the table, the same distance away. That location will always be unoccupied because we maintain the invariant that, following each Alice move, any position on the table is unoccupied if and only iff its a "opposite" position is unoccupied. Since Alice always has a move to make and the table must eventually hold no more pennies, she will win.

Problem 3. *Alice might like to go on a date with Bob. Bob might like to go on a date with Alice. But nobody asks the other out because they're too embarrassed to express interest in case the other is not interested.*

Alice and Bob aim to solve this problem by designing a protocol (an algorithm) in which each learns of the other's interest if both are interested. Said differently, Alice has a private bit $a \in \{0,1\}$ and Bob has a private bit $b \in \{0,1\}$, and we seek a method wherein Alice and Bob can interact with one another and, at the end of the interaction, each will know $a \wedge b$, but nothing more. If $a = b = 1$ they each learn this fact; if $a = 0$, Alice learns nothing of b; if $b = 0$, Bob learns nothing of a.

For your solution, use only simple, physical objects you might find around your home. Assume Alice and Bob are basically honest and cooperative, but don't assume either will do what you say if left unobserved.

There are numerous solutions. Here are three. **Open your eyes.** Alice and Bob agree to the following: "We sit across one another at the table, eyes closed. Now, on the count of three, open your eyes iff you want to go on a date." **Make a date.** Alice and Bob agree to the following: "If you're interested to go on a date, go to the MU coffee shop Friday at 5pm. Otherwise, stay away from there." **Flashlight.** Alice and Bob sit across one another at a table. Each has a D-size battery. Alice takes an old-style flashlight, unscrews the top, and then, under the table, puts in her battery: negative-side down if $a = 1$, positive side-down if $a = 0$. She hands the flashlight to Bob, who, beneath the table, inserts his battery: negative-side down if $b = 1$, positive side-down if $b = 0$. He screws on the top of the flashlight. In view of Alice, he flips the switch. If it fails to light up, he unscrews the top and, beneath the table, dumps out the batteries.

Problem 4. *An n-bit permutation P is a one-to-one and onto function with domain and range $\{0,1\}^n$. The set of all n-bit permutations is denoted* $\mathrm{Perm}(n)$*. By a random n-bit permutation I mean a function drawn uniformly from* $\mathrm{Perm}(n)$*.*

An n-bit cycle C is an n-bit permutation for which $0^n, C(0^n), C(C(0^n)), \ldots, C^{2^n-1}(0^n)$ are distinct. The set of n-bit cycles is denoted $\mathrm{Cycl}(n)$*. By a random n-bit cycle I mean function drawn uniformly from* $\mathrm{Cycl}(n)$*.*

(As a suggested warm-up, draw some pictures illustrative of random permutations and random cycles; figure out why, for a random cycle, $C^{2^n}(0^n) = 0^n$; and compute $|\mathrm{Perm}(n)|$ and $|\mathrm{Cycl}(n)|$.)

Finally, the question: Fix $n \geq 1$. Now show how to convert a random permutation $P \in \mathrm{Perm}(n)$ into a random cycle $C \in \mathrm{Cycl}(n)$. That is, provide a (stateless, deterministic) algorithm to compute $C(x)$ that makes (efficient, black-box) use of permutations $P(y)$ and $P^{-1}(z)$. Explain why C is a cycle, and why it is uniformly random in $\mathrm{Cycl}(n)$ *as long as P is uniformly random in* $\mathrm{Perm}(n)$*.*

Fix n. Let $\mathrm{Inc}\colon \{0,1\}^n \to \{0,1\}^n$ be an arbitrary cycle on $\{0,1\}^n$, say the function that treats its n-bit input as a number, adds 1 modulo 2^n, and then treats the result as an n-bit number. Treating numbers as n-bit strings, $0 \to 1 \to 2 \to \cdots \to 2^n - 1 \to 0$ is the cycle named by Inc. We now just rename each point x on this cycle by $P(x)$. Formally, $C(x) = P^{-1}(\mathrm{Inc}(P(x)))$. Of course you can do it the other way around, too: $C(x) = P(\mathrm{Inc}(P^{-1}(x)))$. Clearly this provides a unifom random cycle if our renaming function P is uniformly random.