1 Practice Problems

1. Create an arithmetic circuit that takes signals x_1, x_2, ..., x_n and is satisfied if *at least* one signal is 0.

   we note that this is effectively the same as saying $\sum_{i=1}^{n} x_i === n-1$

2. Create an arithmetic circuit that takes signals x_1, x_2, ..., x_n and is satisfied if all signals are 1.

$$\sum_{i=1}^{n} x_i - n = 0$$
$$\text{and} \forall i \in [0, n] : x_i(1 - x_i) == 0$$

3. A bipartite graph is a graph that can be colored with two colors such that no two neighboring nodes share the same color. Devise an arithmetic circuit scheme to show you have a valid witness of a 2-coloring of a graph. Hint: the scheme in this tutorial needs to be adjusted before it will work with a 2-coloring. TODO

4. Create an arithmetic circuit that constrains k to be the maximum of x, y, or z. That is, k should be equal to x if x is the maximum value, and same for y and z.

   for two variables we can simulate the max function like so

$$max(x, y) = \frac{x + y + |x - y|}{2}$$

   we can extend this to three variables by using the fact that $max(x, max(y, z)) = max(max(x, y), z)$

$$max(x, max(y, z)) = max(max(x, y), z)$$

   IE we need to satisfy

$$a = \frac{x + y + |x - y|}{2}$$
$$k = \frac{a + z + |a - z|}{2}$$

5. Create an arithmetic circuit that takes signals x_1, x_2, ..., x_n, constrains them to be binary, and outputs 1 if *at least* one of the signals is 1. Hint: this is trickier than it looks. Consider combining what you learned in the first two problems and using the NOT gate.

we note this is the same as saying

$$\sum_{i=0}^{n} x_i \geq 1$$

we can use a combination of the not operator and $\sum_{i=1}^{n} x_i === n - 1$

$$\begin{aligned}
\forall i \in [0, n] : x_i(1 - x_i) &=== 0 \\
a = \sum_{i=1}^{n} x_i &=== 0 \\
res &= 1 - a
\end{aligned}$$

6. Create an arithmetic circuit to determine if a signal v is a power of two (1, 2, 4, 8, etc). Hint: create an arithmetic circuit that constrains another set of signals to encode the binary representation of v, then place additional restrictions on those signals.

   first we constrain v

$$\begin{aligned}
v &= \sum_{i=0}^{n} 2^i x_i \\
\forall i \in [0, n] : x_i(1 - x_i) &=== 0
\end{aligned}$$

   we need an additional constraint that checks that exactly one bit can be set to one

$$\sum_{i=0}^{n} x_i = 1$$

7. Create an arithmetic circuit that models the Subset sum problem (link). Given a set of integers (assume they are all non-negative), determine if there is a subset that sums to a given value k. For example, given the set $\{3, 5, 17, 21\}$ and $k = 22$, there is a subset $\{5, 17\}$ that sums to 22. Of course, a subset sum problem does not necessarily have a solution. Use a "switch" that is 0 or 1 if a number is part of the subset or not.

   we define a set $A = \{a_1, a_2, \ldots, a_n \ : a \geq 0\}$ and a target value k

   we create a switch a_i for each set-element indicating if it is part of the subset.

$$\begin{aligned}
\sum_{i=1}^{n} a_i x_i &=== k \\
\forall i \in [1, n] : x_i(1 - x_i) &=== 0
\end{aligned}$$

8. The covering set problem starts with a set $S = \{1, 2, \ldots, 10\}$ and several well-defined subsets of S, for example: $\{1, 2, 3\}$, $\{3, 5, 7, 9\}$, $\{8, 10\}$, $\{5, 6, 7, 8\}$, $\{2, 4, 6, 8\}$, and asks if we can take at most k subsets of S such that their union is S. In the example problem above, the answer for $k = 4$ is true because we can use $\{1, 2, 3\}$, $\{3, 5, 7, 9\}$, $\{8, 10\}$, $\{2, 4, 6, 8\}$. Note that for each problem, the subsets we can work with are determined at the beginning. We cannot construct the subsets ourselves. If we had been given the subsets $\{1, 2, 3\}$, $\{4, 5\}$ $\{7, 8, 9, 10\}$ then there would be no solution because the number 6 is not in the subsets. On the other hand, if we had been given $S = \{1, 2, 3, 4, 5\}$ and the subsets $\{1\}$, $\{1, 2\}$, $\{3, 4\}$, $\{1, 4, 5\}$ and asked can it be covered with $k = 2$ subsets, then there would be no solution. However, if $k = 3$ then a valid solution would be $\{1, 2\}$, $\{3, 4\}$, $\{1, 4, 5\}$. Our goal is to prove for a given set S and a defined list of subsets of S, if we can pick a set of subsets such that their union is S. Specifically, the question is if we can do it with k or fewer subsets. We wish to prove we know which k (or fewer) subsets to use by encoding the problem as an arithmetic circuit.

   we define a universe $S = \{a_1, a_2, \ldots, a_10\}$
   and N subsets $S_i \subseteq S \forall i \in [1, N]$

   first we can constrain the problem by having a set s which determines if S_i is part of the minimal union solution.

$$\forall i \in [1, N] : s_i(1 - s_i) === 0$$
$$\sum_{i=1}^{N} s_i === k$$

   we now need to find a way to encode union of subsets and avoid duplicates. we can do this by creating additional switches s_{ij} for the jth element in the ith subset a_{ij} specifying if the element is unique in the union.

$$\text{constrain each switch to be binary} \forall i \in [1, N] : \forall j \in [1, |S_i|] : s_{ij}(s_{ij} - 1) === 0$$

$$\text{constrain each element to only be used once} \forall j \in [1, 10] : \sum_{i}^{N} s_{ij} === 1$$

$$\text{constrain the union of all subsets to be the universe} \sum_{i}^{N} \sum_{j}^{|S_i|} s_{ij} * a_i j === \sum_{0}^{10} a_i$$

   this proves that we know a k and subsets that cover the universe but not that we have the minimal k

   we can show we have the minimal k by summing over k-1 subsets and proving that the sum is less than the universe

$$A = \sum_{i}^{k-1} s_{ij} * a_i j === \sum_{0}^{10} a_i \text{apply the not gate} 1 = 1 - A$$