## Warm-Up Exercise

Find a Fermat witness and a root witness for 341, and a Miller-Rabin witness for 561 (the smallest Carmichael number).

## Exercises for Submission

1. Recall that the *greatest common divisor* of $a$ and $b$, denoted $\gcd(a, b)$, is the largest $d$ such that $d|a$ and $d|b$ (where $d|a$ denotes that $d$ divides $a$). Consider the following algorithm:
   $\mathsf{Euclid}(a, b)$: If $b = 0$ return $a$, otherwise return $\mathsf{Euclid}(b, a \bmod b)$.

   (a) Prove that $\gcd(a, b) = \gcd(b, a \bmod b)$, and that $\mathsf{Euclid}(a, b)$ calculates $\gcd(a, b)$.

   (b) Prove that if $a \geq b$ then $a \bmod b < a/2$, and conclude that Euclid can be implemented in polynomial running time (in the *input size*).

   (c) Show that Euclid can be extended to compute two integers $x, y$ such that $x \cdot a + y \cdot b = \gcd(a, b)$. Use it to show a polynomial time algorithm Inverse that given $n$ and $m \in \mathbb{Z}_n^*$ outputs the inverse of $m$ mod $n$ (the unique integer $m^{-1}$ such that $m \cdot m^{-1} = 1 \bmod n$).

2. (a) Show a polynomial-time algorithm that given three positive integers $a, e, n$ outputs $a^e \bmod n$. Hint: Try it first with $e$ which is a power of 2.[1]

   (b) A positive integer $n$ is a *power* if it is of the form $q^k$, where $q, k$ are positive integers and $k > 1$. Show a polynomial-time algorithm for determining whether a positive integer $n$ is a power.

3. (a) Show that $\mathsf{BPP}^{\mathsf{BPP}} = \mathsf{BPP}$.

   (b) Show that if $\mathsf{SAT} \in \mathsf{BPP}$ then $\mathsf{PH} = \mathsf{NP}^{\mathsf{RP}}$.

4. Let PP be the set of languages for which there exists a probabilistic polynomial-time Turing machine $M$, such that for every $x \in L$ the machine $M$ accepts $x$ with probability greater than $1/2$, and for every $x \notin L$ the machine $M$ accepts $x$ with probability at most $1/2$.

   (a) Show that BPP is closed under union and intersection and explain why your argument fails for PP.

   (b) Show that $\mathsf{NP} \subseteq \mathsf{PP} \subseteq \mathsf{PSPACE}$.

5. Let $\mathsf{BPP}_{\mathsf{path}}$ be the class of all languages $L$ that can be decided by a polynomial-time probabilistic Turing machine with the following properties: For every $x \in L$ at least $\frac{2}{3}$ of the computation paths end with a 'yes'. For every $x \notin L$ at least $\frac{2}{3}$ of the computation paths end with a 'no'. Prove that $\mathsf{NP} \subseteq \mathsf{BPP}_{\mathsf{path}}$.[2]

---

[1] Analyze the running time in terms of number of multiplications. Be specific about the number of bits each number you store takes.

[2] A computation path is determined by a sequence of coins. In the standard BPP class, for every input in the language the probability to accept is at least $\frac{2}{3}$, but it does not imply that at least $\frac{2}{3}$ of the computation paths end with 'yes', since they might have different probabilities.