1. (a) Let $L \in \mathsf{NP}$. By definition, there is a predicate $R(x, y)$ computable in $O(poly(|x|, |y|))$ time and $k \geq 1$ such that

$$(\forall x)[x \in L \iff (\forall x)(\exists y : |y| \leq |x|^k) R(x, y)].$$

Thus an interactive proof system for L is given by the following protocol.

On instance x,

Prover: Send a proof y of x.

Verifier: Accept if and only if $R(x, y)$.

Clearly this runs in polynomial time; a prover that send y will always succeed, and a prover that does not send y will always fail. Hence $L \in \mathsf{IP}$.

(b) Let $L \in \mathsf{IP}$, and let V be a corresponding prover and verifier for L. Let x be an input string. The key observation is that the history H_x of the conversation between the best possible prover and V (the messages between them) is a string of length $poly(|x|)$.

Let a possible conversation history of messages be $m_1, \ldots, m_{p(|x|)}$ for some polynomial p where odd-numbered ones are messages from the prover, and even-numbered ones are messages from the verifier. Each such history causes V to either accept or reject.

First, represent all possible conversations by the prover and verifier as a complete tree of $p(|x|)$ depth, where each inner node has $2^{poly(|x|)}$ children. (Call this the *conversation tree.*) Each edge from a parent to a child corresponds to a possible message m_i. Odd-numbered levels of the tree represent points where messages are sent from prover to verifier, and even-numbered levels represent messages from verifier to prover. (Note this tree is of size $2^{poly(|x|)}$.) Thus a path from the root to a leaf represents one possible conversation history. A leaf is labelled either *accept* or *reject*, depending on what the conversation history causes the verifier to do. Since the prover is unbounded, odd-numbered messages are chosen to maximize the acceptance probability of the verifier, given the previous messages. The even-numbered ones are dependent on the private random string of V and the previous messages. Denote the probability of message m_k by $P(k)$.

For a node v, let's define $P(v)$ to be the probability that the *best possible prover* makes V accept, if the protocol is executed starting from node v. (More precisely, it's the maximum probability over all provers that V accepts, when the past conversation history is given by the edges on the path from the root to v.) For accept nodes, $P(v) = 1$, and for reject nodes, $P(v) = 0$. For inner nodes, we can determine $P(v)$ in a bottom-up fashion:

- If the level of v is odd, $P(v)$ is the *maximum* $P(v')$, for all children v' of v.
- If the level of v is even, let $p_{v'}$ be the probability that the verifier chooses the message given by edge (v, v') to send, when the conversation history is the path from the root

of the tree to v. Then $P(v)$ is $\sum_{v'}(p_{v'} \cdot P(v'))$, where the sum is over all children v' of v.

Let r be the root of the conversation tree. Our goal is to compute the probability $P(r)$. Knowing this immediately determines if the interactive proof system accepts or not.

We argue that $P(r)$ can be determined in polynomial space. The key idea is to use depth-first search to compute the $P(v)$'s. We define a procedure ComputeP(v,i) that returns $P(v)$ for v on level i:

ComputeP(v,i):
  If $i = p(|x|)$, return the accept/reject behavior of V on the conversation history given by the path from r to v.
  Set $C_i := 0$.
  For all children v' of v,
    Set $D :=$ ComputeP$(v', i+1)$.
    If i is even, set $C_i := C_i + (p_{v'} \cdot D)$.
    If i is odd, if $(D > C_v)$ then set $C_v := D$.
  End for.

Finally, we claim that ComputeP$(r, 1)$ can be evaluated in polynomial space. Clearly, each update to a C_i can be done in polynomial time, given the proper D. We only need extra workspace to store D, the current path from a node v to the root, the strings on that path's edges, and each counter $C_1, \ldots, C_i$ created along this path. But each C_i is of at most polynomial size, since each $p_{v'}$ and $P(v)$ take a polynomial number of bits to describe.

2. (a) The protocol for V_k is:
  Repeat $|x|^k$ times:
    If $(P_k \leftrightarrow V)(x)$ accepts, then return *accept*.
  End repeat.
  Return *reject*.
  That is, V_k simulates V for a polynomial number of times.
  Clearly, if $\Pr[(P_k \leftrightarrow V)(x) \text{ accepts}] = 1$, then the probability that the above protocol accepts is 1. (Thus a prover P_k that just repeats the behavior of P will always convince the verifier.) For all provers P_k, if $\Pr[(P_k \leftrightarrow V)(x) \text{ accepts}] \leq 1/2$, then the probability that the above protocol accepts is at most $1/2^{|x|^k}$, as each run of the protocol is independent. Hence the above prover P_k and verifier V_k have the desired properties.

(b) Let $d > 1$ be a constant to set later. The protocol for V_k is:
  $C := 0$.
  Repeat $d|x|^k$ times:
    If $(P_k \leftrightarrow V)(x)$ accepts, then increment C.
  End repeat.

Return *accept* iff $C > d|x|^k/2$.

That is, V_k simulates V and takes the majority of outcomes. We consider two cases.

- If $\Pr[(P_k \leftrightarrow V)(x) \text{ accepts}] > 2/3$, then the probability that the above protocol accepts is the probability that the sum of $d|x|^k$ independent random variables $X_1 + \cdots + X_{24|x|^k}$ exceeds $d|x|^k/2$, where $\Pr[X_i = 1] = 2/3$, $\Pr[X_i = 0] = 1/3$. This probability is
$$1 - \Pr[X_1 + \cdots + X_{d|x|^k} \leq \mu - \frac{d|x|^k}{6}],$$
where $\mu = E[X_1 + \cdots + X_{10|x|^k}] = 2/3 \cdot (d|x|^k) = \frac{2}{3}d|x|^k$. By a Chernoff bound, this is
$$1 - \Pr[X_1 + \cdots + X_{d|x|^k} \leq (1 - 1/2)\mu] \geq 1 - e^{-\frac{(1/2)^2\mu}{2}} = 1 - e^{-\frac{d|x|^k}{16}}.$$
Setting $d \geq 16$ ensures that this probability is sufficiently high.
- If $\Pr[(P_k \leftrightarrow V)(x) \text{ accepts}] < 1/3$, then the above setup changes with $\Pr[X_i = 1] = 1/3$, $\Pr[X_i = 0] = 2/3$, $\mu = (1/3) \cdot |x|^k$. The probability of acceptance is
$$\Pr[X_1 + \cdots + X_{d|x|^k} > \mu + \frac{d|x|^k}{6}],$$
which by Chernoff bounds is
$$\Pr[X_1 + \cdots + X_{d|x|^k} > (1 + 1/2)\mu] \leq e^{-\frac{(1/2)^2\mu}{3}} = e^{-\frac{d|x|^k}{24}}.$$
Setting $d \geq 24$ ensures a sufficiently low probability of acceptance.

3. We denote the kth integer in the continued fraction expansion of a number n by $a(n,k)$, starting with $k = 0$.

   (a) First, $a(e, 0) = 2$.
   When $k = 3\ell - 1$ for some integer ℓ, then $a(e, k) = 1$.
   Otherwise, $a(e, k) = 2k$ for $k > 0$.

   (b) $a(\phi, k) = 1$, for all k.

   (c) $a(\tan(1), k) = 1$ if k is even, and $a(\tan(1), k) = k$ if k is odd.

   (d) **(Extra Credit)**
   $$1/(1 + 1/(2 + 1/(3 + 1/(4 + \cdots)))) = I_1(2)/I_0(2) \approx 0.697774,$$
   where $I_n(k)$ is the modified Bessel function of the first kind.

   References: Mathworld
   `http://mathworld.wolfram.com/ModifiedBesselFunctionoftheFirstKind.html`,
   and Sloan's Encyclopedia of Integer Sequences
   `http://www.research.att.com/~njas/sequences/A052119` .

4. We follow the approach of Lecture 7.

(a) Consider the continued fraction expansion

$$0.141592 = 0 + 1/(7 + 1/(15 + 1/(84 + 1/(6 + \cdots)))).$$

The sequence of approximations to 0.141592 reads:

$$\frac{1}{7}, \frac{15}{106}, \frac{16}{113}, \frac{1369}{9598}, \ldots.$$

From there, the numerator and denominators (in lowest possible terms) are only increasing. But 113 is a three-digit prime, and thus a candidate for p such that $1/p = 0.00 \cdots 141592 \cdots$.

(b) By way of Maple, we obtain

$$1/113 = \cdots\cdots 8\ 141592\ 9 \cdots ,$$

which occurs somewhere north of the 70th digit in the decimal expansion.

5. This was basically a freebie.

6. Omitted (for now).