Here is a brief outline of the construction I did in class last Friday to prove that there is a decision problem that is decidable but not in P.

We will build a TM M that halts on all inputs but differs from every machine that runs in polynomial time. The technique is called *clocked diagonalization*. On input x, M takes the following steps.

1. Check that x is of the form $N\#0^m$, where N is a valid TM description. Reject if not.
2. If $x = N\#0^m$, let $n = |x|$. Call a subroutine to lay off $f(n)$ tape cells on a separate track of the tape, where $f(n)$ is any computable superpolynomial function (that is, $\Omega(n^k)$ for all k). In class I did a Fibonacci trick to lay off φ^n tape cells, where φ is the golden ratio $(1+\sqrt{5})/2 \approx 1.618\cdots$, but you could just as easily use 2^n, 3^n, or any other superpolynomial function computable by a Turing machine.
3. Simulate N on input x with a universal machine for up to $f(n)$ (simulated) steps, counting on a separate track. If N halts within that time, halt and do the opposite: if N rejects, accept; and if N accepts, reject. If N does not halt within that time, halt and accept.

Then M halts on all inputs. Now we claim that $L(M) \neq L(N)$ for all Turing machines N running in polynomial time. Let N be an arbitrary machine running in polynomial time, say n^k on all but perhaps finitely many inputs. Let n_0 be large enough that N runs in time n^k for all $n \geq n_0$ and $f(n) \geq n^k$ for all $n \geq n_0$. Let m be large enough that $|N\#0^m| \geq n_0$. Then on input $x = N\#0^m$, M will lay off $f(n)$ tape cells and simulate N on input x for at most $f(n)$ steps, and the simulation will have time to complete. Since N halts within the time limit, M will do the opposite of what N does on input x. Thus $L(M)$ and $L(N)$ differ on x. Since N was an arbitrary polynomial-time machine, $L(M)$ is not accepted by any Turing machine running in polynomial time.