Here is a possible solution to Problem 2 on HW6: finding a max-cut using randomness. Most people gave a greedy algorithm for solving the problem, which received full credit but kind of missed the point. Two groups came up with the following (Mike Picollelli and Kelley Burgin, Haowen Chan and Yan Ke), which is what I initially intended. Sorry if it was ambiguous.

Algorithm: Choose a random cut (i.e. a random partition of the vertices into two sets). If there are $< m/2$ edges crossing this cut, repeat.

Analysis: Let X be a random variable denoting the number of edges crossing a cut. For $i = 1, \ldots, m$, X_i will be an indicator variable that is 1 if the ith edge crosses the cut, and 0 otherwise. Then $X = \sum_{i=1}^{m} X_i$, so by linearity of expectation, $E[X] = \sum_{i=1}^{m} E[X_i]$.

Now for any edge $\{u, v\}$, the probability it crosses a randomly chosen cut is 1/2. [Why? We randomly placed u and v in one of two possible partitions, so u is in the same partition as v with probability 1/2.] Thus, $E[X_i] = 1/2$ for all i, so $E[X] = m/2$.

This only shows that by choosing a random cut, we expect to get at least $m/2$ edges crossing. We want a randomized algorithm that *always* returns a good cut. Let's compute the probability that $X \geq m/2$ when a random cut is chosen. In the worst case, when $X \geq m/2$ all of the probability is weighted on m, and when $X < m/2$ all of the probability is weighted on $m/2 - 1$. This makes the expectation of X as high as possible, while making the likelihood of obtaining a $\geq m/2$ cut small. Formally,

$E[X] \leq (1 - Pr[X \geq m/2])(m/2 - 1) + Pr[X \geq m/2]m.$

But $E[X] = m/2$ from the above. Solving for $Pr[X \geq m/2]$, it is at least $\frac{2}{m+2}$. It follows that the expected number of iterations in the above algorithm is at most $(m+2)/2$; therefore the algorithm runs in expected polynomial time and always returns a cut of size $\geq m/2$.