Online bribery

In the online bribery problem, there is a hidden value, a positive integer x, and you ask a sequence of questions $q_1, q_2, \ldots$ until you find one that is greater than x. What you pay is the sum of your bids, $\sum_i q_i$. If you knew everything ahead of time, you would know x and simply ask x. Thus the competitive ratio objective is to design an algorithm minimizing $\sum q_i/x$.

The doubling algorithm asks the sequence of questions $q_i = 2^i$.

Theorem 1 *The doubling algorithm is a 2-approximation. This is optimal for deterministic algorithms.*

If you stop with q_n, then your cost is about $2q_n$ and x is at least $q_{n-1} = q_n/2$, hence the 4-approximation.

For the lower bound, consider any sequence and assume that it's an a approximation for $a < 4$. Let $s_n = \sum q_i$, $y_n = s_{n+1}/s_n$. If the adversary picks x just above q_n, we must have $s_{n+1}/q_n < a$ for all n. Rewrite, do the algebra, and deduce that at some point $s_{n+1} < s_n$, a contradiction. Hence the theorem.

Here is a randomized algorithm: pick a random number u drawn uniformly in $[0, 1]$, and let $q_i = 2^{i+u}$ (rounded). To analyze it, observe that for any x, the competitive ratio will be $2q_n/x$. But the fractional part of $\log_2(q_i/x)$ is independent of i, so even for the final question, it is distributed as the fractional part of $\log_2(q_0/x)$, namely, uniformly in $[0, 1]$. So in expectation it q_n/x is $\int_0^1 2^u du$. This determines the competitive ratio.

Exercise (due Wednesday): do the same using the natural logarithm, i.e. $q_i = e^{i+u}$. What is the competitive ratio of this randomized algorithm?