Hunting Hoglins in Hogwarts

Problem author and developer: Tikhon Evteev

The main idea is to not do the binary search. If you try, you will probably end up with about 400 caught hoglins, which is not enough.

Instead we will maintain the set of disjoint segments $s_1, \ldots, s_k$ of approximately equal length. i.e. $||s_i| - |s_j|| \leq 1$. After each iteration of the algorithm, it will hold that the hoglin is in one of those segments, and this segment is also the segment that the hoglin considers accessible. We will also keep track of the probabilities that the hoglin is in each of these segment, and denote them as p_i. We will reorder the segments so that $p_1 \geq p_2 \geq \cdots \geq p_k$.

From this point on we will not keep track of the fact that the segments are of approximately equal length, and write as if they were of exactly equal length. We will also not mention, that if at any point of the interaction we catch the hoglin, we should immediately stop and restart the process.

Initially the hoglin is in a segment $s_1 = [1, n]$ with the probability $p_1 = 1$.

For the iteration of the algorithm we will go through the segments in order from 1 to k and block the midpoint of the corresponding segment. If the interactor returns 1 after we block the midpoint of the j-th segment, than the hoglin was in one of the segments $s_1, \ldots s_j$ and now is in one of the $2 \cdot j$ new segments, each of which is approximately two times smaller than the segments in the previous iteration. If we don't get a 1 after we block the last midpoint, we will just wait for it by sending 0, and let $j = k$.

Let's for now denote these segments as $s_1^1, s_1^2, s_2^1, s_2^2, \ldots, s_j^1, s_j^2$, and compute the probabilities $p_1^1, p_1^2, p_2^1, p_2^2, \ldots, p_j^1, p_j^2$ with which the hoglin is in each of these segments.

The conditional probability that we will get a 1 after blocking the j-th midpoint, if the hoglin initially was in the i-th segment is $P(R_j|W_i) = \frac{1}{2^{j-i+1}}$, thus the probability that it is now in either of the segments s_i^1 or s_i^2 by Bayes' theorem is $p_i^1 + p_i^2 = P(W_i) = \frac{p_i \cdot P(R_i|W_i)}{P(R_i)} = \frac{p_i \cdot P(R_i|W_i)}{\sum_{l=1}^{j} p_l \cdot P(R_l|W_l)}$.

So we can recompute the probabilities p_i for the next step and continue the algorithm.

Now to prove the asymptotics we can prove by induction that the sequence $p_1, \ldots p_k$ looks like $p, \ldots p, \frac{p}{2}, \ldots \frac{p}{2}, \frac{p}{4}, \ldots \frac{p}{4}, \ldots$. More over, each sequence of the form $\frac{p}{2^l}, \ldots \frac{p}{2^l}$ is at most 4 elements long.

To show that we will use the fact that $p_i^1 = p_i^2 \propto p_i \cdot P(R_i|W_i) = \frac{p_i}{2^{j-i}}$. Not strictly: the sequence of probabilities is multiplied by the sequence 2^i, and then each one is duplicated and the sequense is normalized. Thus the sequence on the equal probabilities is no longer than 4, and the probabilities are increasing exponentially.

Knowing that, we do not need to maintain the probabilities p_i to sort the segments in the right order, because we can just reverse the order of segments after each iteration and it will maintain the right order.

Also knowing the exponentially decreasing nature of p_i we can see that the expected value of the number of queries for each iteration is at most $1 \cdot p + 2 \cdot p + 3 \cdot p + 4 \cdot p + 5 \cdot \frac{p}{2} + 6 \cdot \frac{p}{2} + \ldots$, where $p = \frac{1}{8}$, which is $\frac{5}{2} + 4 = 6\frac{1}{2}$. Since there are at most $\log_2 n$ iteration, the solution works in at most $6\frac{1}{2} \cdot \log_2 n$ queries per hoglin on average.

The problem demands no more then about $4.2 \cdot \log_2 n$ queries per hoglin, but one can see how our analysis was only a rough upper bound on the number of queries. In reality, the solution works in about $3.9 \cdot \log_2 n$ queries per hoglin, which leaves more then enough room for variance.