100 Prisoners and A light Bulb

Yisong Song

1. Introduction

The article represent three different protocols for solving the “100 Prisoners and a Light Bulb” riddle, including the explicit computations of average runtime. The article also discusses the variation of the original riddle and explore the existence of the solution in altered situation.

2. The riddle

The riddle is as followings:

One hundred prisoners have been newly ushered into prison. The warden tells them that starting tomorrow, each of them will be placed in an isolated cell, unable to communicate amongst each other. Each day, the warden will choose one of the prisoners uniformly at random with replacement, and place him in a central interrogation room containing only a light bulb with a toggle switch. The prisoner will be able to observe the current state of the light bulb. If he wishes, he can toggle the light bulb. He also has the option of announcing that he believes all prisoners have visited the interrogation room at some point in time. If this announcement is true, then all prisoners are set free, but if it is false, all prisoners are executed. The warden leaves, and the prisoners huddle together to discuss their fate. Can they agree on a protocol that will guarantee their freedom?

3. Protocol Design

3.1 Basic Assumption

In order to develop a feasible strategy for the problem, we firstly make some basic assumption to the ambiguous situation described above. In next session, we will discuss if we can construct a solution without these assumption.

Assumption 1 Prisoners can count how may days have elapsed.
Assumption 2 The initial bulb state is OFF.

3.2 Protocol I

Since warden chooses prisoner uniformly, each prisoner will eventually get into the room once. Based on this fact we can develop our first solution.

3.2.1 Strategy: The days are split into n-day blocks. During each n-day block, each prisoner operates according to the following instructions upon entering the interrogation room:

- If it is day 1 for the current block:
    - If the bulb is OFF, turn the bulb ON.
    - If the bulb is already ON, and the first n-day block has already elapsed, announce that all prisoners have visited.

- On any other day of the current block:
    - If it is your first time visiting the room during the current block, do nothing.

- If it is your second time visiting the room during the current block, turn the light OFF.
- If it is your third or more time visiting the room during the current block, do nothing.

3.2.2 Explanation: The general idea is that eventually, with probability 1, we will be lucky enough to have a block of n-days during which no prisoner enters the room twice, or in other words, during which every prisoner will enter the room exactly once. Then the bulb which was turned ON on day 1 will still be ON after n-days, since bulbs are only turned OFF upon a second return visit. Thus, if the bulb remains ON on the first day of a new block, we know that every prisoner must have visited the interrogation room during the block that had just elapsed.

3.3.3 Runtime Evaluation: We now compute the expected runtime of this protocol. Let X be the number of days the protocol requires. Let B be the number of n-day blocks required till the protocol succeeds. Then B is a geometric random variable with parameter

$$\frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \quad \frac{n-3}{n} \quad .... \quad \frac{1}{n} = \frac{n!}{n^n}$$

Since the expectation of a geometric random variable is the reciprocal of its parameter, and X = $n \times B$, the expected number of days required is

$$\boldsymbol{E}[\mathrm{X}] = n\boldsymbol{E}[\mathrm{B}] = n\frac{n^n}{n!} = \frac{n^{(n+1)}}{n!}$$

Using Stirling’s approximation n! ~ $\sqrt{2\pi n}(\frac{n}{e})^n$, we can get the conclusion that

$$\mathbf{E}[\mathrm{X}] \sim \frac{1}{\sqrt{2\pi}} n^{1/2} e^n = O(n^{1/2} e^n).$$

When n = 100, **E** [X] equals 1.072 × 10^{44} days. Statically speaking, this can't be taken as an acceptable solution.

3.3 Assumption on asymmetric

One of the possible sources of difficulty in solving this riddle is the natural idea that every prisoner should should follow the same instructions. If we can make the assumption that each prisoner can follow different instruction, we can make the problem easier to solve.

Assumption 3 : The protocol can be asymmetric, which means each prisoner can follow different instruction.

3.4 Protocol II (One Counter Strategy)

3.4.1 Strategy: Letting prisoners have different roles, we assign one prisoner to be “the counter”. He will maintain an integer variable in his head that is initialized to 1. Call this variable T . Upon entering the room, prisoners adhere to the following instructions:

- If you are not the counter:
  - If the bulb is OFF, and you have never turned the bulb ON before, turn it ON.

  - If the bulb is ON, do nothing.

- If you are the counter:
  - If the bulb is OFF, do nothing.
  - If the bulb is ON, turn it OFF, and set T=T+1.
  - If T = n, announce that all prisoners have visited.

3.4.2 Explanation: The idea behind this protocol is that every prisoner besides the counter will turn ON the bulb exactly once, whenever he can. When the bulb is ON, no one can turn it OFF except for the counter. Eventually the counter will enter the room, turn this bulb OFF, and increment the count T . In this way, each prisoner indicates his presence in the room to the counter by leaving an ON bulb which is eventually recorded by the counter.

3.4.3 Runtime Evaluation: To analyze the runtime, we can split the process into epochs. Let X_i denote the number of days between the first day on which T = i, and the first day on which T = i + 1. Between these two days, two events must occur:

(1) An unrecorded prisoner must be chosen, causing the bulb to be turned ON. Let Y_i denote the number of days between from when T = i until this event occurs.

(2) The counter must then enter the room to record this ON bulb. Let Z_i denote the number of days from when the bulb is turned ON until this occurs.

Then

$$X_i = Y_i + Z_i$$

Letting X be a random variable corresponding to the number of days the protocol requires in total, we have

$$X = \sum_{i=1}^{n-1} X_i = \sum_{i=1}^{n-1} Y_i + Z_i$$

Y_i is a geometric random variable with parameter $\frac{n-i}{n}$, and Z_i is a geometric random variable with parameter $\frac{1}{n}$, Hence, by linearity of expectation, the expected runtime is

$$\begin{aligned} \boldsymbol{E}[\mathrm{X}] &= \sum_{i=1}^{n-1} E[Y_i] + E[Z_i] \\ &= \sum_{i=1}^{n-1} \left(\frac{n}{n-i} + n\right) \\ &= (n-1)n + n\sum_{i=1}^{n-1} \frac{1}{i} \\ &= n^2 - n + nH_{n-1} \end{aligned}$$

In big O, since $H_n \sim \ln n$,

$$\boldsymbol{E}[\mathrm{X}] = O(n^2)$$

When n = 100, E [X] equals 10417.74 days, or 28.54 years, which is an acceptable answer.

The variance of this protocol is also be easily computed. Asymptotically,

$$\boldsymbol{var}[\mathrm{X}]=O(\ n^3\ ).$$

3.4.4 Improvement :

We can make some improvement to the Single Counter Protocol in different aspect.

Improvement.1: Under the one counter protocol, the prisoners escape if and only if the bulb, which is initially OFF, alternates its state from OFF to ON exactly n−1 times. Non-counters can also count these state transitions as they witness them. So, a marginal improvement in the algorithm can be made by realizing that if any very lucky non-counter witnesses all n − 1 such transitions before the counter does, then the non-counter is equally qualified to declare victory and preempt the counter in the very last epoch of the algorithm.

However, Since the standard one counter protocol already requires a runtime of O(n^2), and this new policy for non-counters can only save at most n days (since it only affects the last epoch), the improvement does not affect the asymptotic.

Improvement.2: The One Counter Protocol can be slightly improved by assigning the role of counter dynamically, rather than a priori. We use the following policy: the counter is the first person to enter the room twice in the first n days.

- Stage I: Days 1 through n:
  - Days 1 through n − 1: The first person to enter the room twice will turn the bulb ON, and assign himself to be the counter.
  - Day n: If the light is still OFF, declare victory. Otherwise, turn off the light.

- Stage II: (all remaining days)
  Follow the normal One Counter Protocol, but with the following modifications:
  - The counter only counts up to n − k + 1, where k is the index of the day that the counter entered the interrogation room twice.
  - Prisoners who saw an ON bulb in Stage I do nothing.

To illustrate the idea behind this protocol, suppose we have 100 prisoners, and the first person to enter the interrogation room twice enters on day 20. This prisoner becomes the counter, and he can deduce that in the previous 19 days, there have been exactly 19 distinct visitors, including himself. Thus, when Stage II ensues, he would only need to tally (n-1) - (k-2) = n-k+1 = 99-18 = 81 prisoners. Lastly, if we are so lucky that no counter is assigned on the 100th day, then every visitor in the first 100 days must have been distinct , so we declare victory.

The dynamic counter assignment does constitute an improvement in average runtime over the One Counter Strategy. When n ≥ 4, we have the conclusion that

$$\boldsymbol{E}[\mathrm{X}^{(\text{single counter})}]\ -\ \boldsymbol{E}[\mathrm{X}^{(\text{dynamic assignment})}]\ \geq 1/8$$

Improvement.3: Observe that in the Single Counter Protocol, we will have long stretches of time where the bulb is ON and we are waiting for the counter to enter the room. This suggests that it maybe useful to have multiple alternative counters who are also authorized to record the ON bulb and turn it OFF. Furthermore, it would be nice if we could count faster to n. That is, rather than counting 1-by-1 to n, what if we counted in jumps of 10 instead?

The following Two-Stage Counting Protocol improves on the Single Counter Protocol in both of the aforementioned aspects. Firstly, it divides up the task of counting the prisoners amongst a group of assistant counters. Secondly, the head counter counts up to n more quickly by collecting the aggregated counts of the assistant counters.

3.5 Improved Protocol (Two Stage Counting)

3.5.1 Strategy: To begin the protocol's description, there are three different possible roles for a prisoner: head counter, assistant counter, and "drone". There is exactly one head counter, and there is some number of assistant counters $a \ll n$, while the vast majority prisoners are still drones regular prisoner with no counting tasks. The head counter and all assistant counters all have an integer variable in their heads, initialized to one.

The protocol has two stages, Stage I and Stage II. Each stage lasts for a certain number of preset days, which we will call s_1 and s_2 , respectively. In Stage I, each assistant counter is responsible for counting a quota of q drones. In Stage II, the head counter will be responsible for counting up the assistant counters who have reached their quota. In this way, the head counter counts toward n in jumps of size q. If the head counter does not succeed by the end of Stage II, then we repeat Stage I and Stage II again, still maintaining all the mental counts from before. In other words, we repeatedly alternate between Stages I and II until victory is declared.

Runtime Evaluation: the average runtime of this algorithm is difficult to compute, and remains open for now. However, simulations with certain parameters for the case of n = 100 yield runtime between 3500 and 4000 days, or 9.5 to 11 years.

3.6 Protocol III (Binary Tokens)

The basic idea behind the two stage counting protocol was that to speed things up, sometimes we should count in clumps rather than one-by-one. In the first stage, assistant counters counted one-by-one, and the second stage, the master counter counted the clumps collected by the assistant counters.

This same protocol can be thought of in terms of exchanging "tokens" with variable point values. To make the analogy clear, imagine that all prisoners not assigned any counting roles start with a token worth one point. During Stage 1, these prisoners try to deposit their one-point tokens into the central room by turning on the bulb when they can, and assistant counters collect the tokens. Suppose assistant counters are ordered to count up to 10. Then in Stage 2, assistant counters exchange their collected tokens with 10-point tokens, and try to deposit these 10-point tokens into the room by turning on the bulb when they can. The master counter collects these bigger tokens. Thus, a lighted bulb represents a different number of points depending on what stage we are in, and the prisoners can escape more quickly by counting in terms of gradually higher denomination tokens.

3.6.1 Strategy: The "binary tokens scheme" is a generalization of these ideas. The value of a lighted bulb is doubled from stage to stage, and all prisoners now have the same role, allowed both to deposit points and collect points. Proceeding formally, let n be the total number of prisoners, and suppose n is a power of 2. Let P_k be the number of points a lighted bulb is worth on day k. We will define it later, but for now, know that every P_k is a nonnegative power of 2.

All prisoners use the following instructions:

- Keep an integer in your head; call it T . Initialize it to T = 1.
- Let T_m denote the mth bit of T expressed in binary.
- Upon entering the room on day k, where P_k = 2m for some m, go through four steps:

(1) If the bulb is ON, set $T := T + P_k - 1$, and turn it OFF.
(2) If $T \geq n$, declare victory.
(3) If $T_m = 1$, turn the bulb ON, and set $T := T - P_k$
(4) Else, if $T_m = 0$, leave the bulb OFF and do nothing.

Notice that Step 1 amounts to taking a token worth P_{k-1} points left over from the previous day, and Step 2 amounts to depositing a token worth P_k points. In short, all prisoners will collect and deposit tokens whenever they may legally do so, where the value of tokens are universally dictated by a prespecified sequence P_k that is only a function of what day it is. Whenever someone accumulates 100 points worth of tokens, the game is over.

It remains to specify what P_k should be. The sequence should start with a block of consecutive ones, since everyone starts with only one point. If this block is long enough, there will be many prisoners who have collect more than one point, and perhaps a subsequent block of twos would be effective. We choose the nondecreasing sequence

$$\{ \mathbf{P_k} \} = (\underbrace{1,1,\ldots 1}_{n\ln n + n\ln\ln n} , \underbrace{2,2,\ldots 2}_{n\ln n + n\ln\ln n} , \underbrace{4,4,\ldots 4}_{n\ln n + n\ln\ln n} ,\ldots, \underbrace{\frac{n}{2},\frac{n}{2},\ldots\frac{n}{2}}_{n\ln n + n\ln\ln n})$$

where $T := \log_2 n(n\ln n + n\ln\ln n)$, the length of the finite sequence on the right-hand side. There are $\log_2 n$ stages, each lasting $n\ln n + n\ln\ln n$ days (rounded). In the k^{th} stage, the bulb is worth 2^k , where k indexes from 0 to $(\log_2 n) - 1$.

Lastly, if victory has not been declared after T days, the prisoners will maintain the integers in their heads, and (P_k) restarts. That is, the full sequence (P_k) is T -periodic:

$$P_k := 2^m \quad \text{where} \quad m := [\frac{k \, mod(T)}{n\ln n + n\ln\ln n}]$$

3.6.2 Runtime Evaluation: Firstly let us work through a simple example. Suppose we have n = 4 prisoners labeled A, B, C, and D. Stage 0, in which the bulb is always worth 1 point, then lasts for $\lceil n\ln n + n\ln\ln n \rceil$ days. In the beginning, every prisoner starts with one point, and the bulb is OFF. We can represent this initial state by the table

Day 0

OFF	2^1	2^0
A	0	1
B	0	1
C	0	1
D	0	1

Table 3.6.2.1

where the bulb's status is indicated in the upper left, and the integers being mentally maintained by each of the prisoners is listed in binary in the lower right. Let us play out the following sequence of visitations in Stage 0: A, B, C, B, A, . . . , D.

On Day 1, A is chosen. Following the protocol, A will turn the bulb ON and decrement his number. The new state becomes:

End of Day 1

OFF	2^1	2^0
A	0	0
B	0	1
C	0	1
D	0	1

Table 3.6.2.2

On Day 2, B is chosen. He sees the ON bulb, turns it off, and increments his count. He then checks if the zeroth bit of his newly incremented count is a 1, but it is not, so he does not activate the bulb. The new state is:

End of Day 2

OFF	2^1	2^0
A	0	0
B	1	0
C	0	1
D	0	1

Table 3.6.2.3

On Day 3, C is chosen. This leads to:

End of Day 3

OFF	2^1	2^0
A	0	0
B	1	0
C	0	0
D	0	1

Table 3.6.2.4

On Day 4, suppose that B is chosen again. B sees the bulb, still worth 1 point, and turns it OFF. He then increments his count to 2 + 1 = 3, which is 11_2 in binary. Then he sees that the zeroth bit of his count so far is a 1, so he decrements his count back to 2, and turns the bulb ON again. So within Day 4, we have

End of Day 4

OFF	2^1	2^0
A	0	0
B	1	1
C	0	0
D	0	1

Table 3.6.2.5

To End of Day 4

OFF	2^1	2^0
A	0	0
B	1	0
C	0	0
D	0	1

Table 3.6.2.6

which is the same state as the previous day. In short, choosing B again has no effect on the system.

Now suppose that on Day 5, A (or equivalently, C) is chosen. The consequent behavior will again be identical to that of B on Day 4. Any prisoner with a zeroed count will add and then immediately subtract out whatever the bulb is worth on that day to his count, resulting in no net state change. Thus, any prisoner whose count reaches zero can be thought of as being inactive for the rest of this stage.

Hence, we see that in the remaining days of Stage 0, no net state change will occur unless D, the only person unchosen so far, is chosen, which would lead to the last state in Stage 0:

OFF	2^1	2^0
A	0	0
B	1	0
C	0	0
D	0	1

Table 3.6.2.7

Change to

OFF	2^1	2^0
A	0	0
B	1	0
C	0	0
D	1	0

Table 3.6.2.8

Notice all ones have been paired into groups of two. Stage 1 then proceeds much like Stage 0 did, except that now we increment/decrement starting with the left column of bits, and the number of active prisoners has been halved from four to two. It is easy to see where this binary pattern is going; at the start of Stage k, we should have combined all the 2^{k-1} tokens into 2^k tokens, and there should be only n/ 2^k active prisoners left.

When does the protocol fail? Notice that if D is never chosen in Stage 0, he will never have another chance to deposit his 1-point token into the room since the value of the bulb only goes up in future stages. Thus, the only way the protocol could succeed in this cycle (going through all stages once) is if D is the victory-declaring prisoner which collects all n points in the end. In general though, if there are ever even just two prisoners who are not chosen in a stage, this entire cycle is destined to fail. So, we can draw the following conclusion:

Up to a negligible fencepost error, the binary tokens protocol succeeds if and only if in each stage, every active prisoner is chosen at least once, where the number of active prisoners in Stage k is n/2^k^.

Thus in the k th stage, we collect n/ 2^k tokens, and we have $n\ln n+n\ln\ln n$ days to do it. If **P**[$F_j^{(k)}$] is the probability of failing to collect the j^th^ token at the k^th^ stage, where j ∈ {1, . . . , n/ 2^k }, then

$$\begin{aligned}
\mathbf{P}[\ F_j^{(k)}] &= \left(1-\frac{1}{n/2^k}\right)^{n\ln n+n\ln\ln n} \\
&= \left(e^{-2^k}\right)^{\ln n+\ln\ln n} \qquad \text{as } n \to \infty \\
&= \left(e^{\ln n+\ln\ln n}\right)^{-2^k} \\
&= (n\ln n)^{-2^k} \\
&\leq \frac{1}{n\ln n}
\end{aligned}$$

Then, by invoking the union bound, **P**[$F^{(k)}$] , the probability of the k^th^ stage failing, is

$$\mathbf{P}[\ F^{(k)}] \ \leq \ \sum_{j=1}^{n/2^k} P\ [\ F_j^{(k)}] \leq \ \frac{1}{2^k} \quad \frac{1}{\ln(n)} \ .$$

Let F denote the event that one cycle of the protocol fails. Since the protocol fails if and only if at least one of the $\log_2$ n stages fails, we can again invoke the union bound:

$$\mathbf{P}[F] \leq \sum_{k=0}^{\log_2 n-1} P\ [\ F^{(k)}] \leq \sum_{k=0}^{\log_2 n-1} \frac{1}{2^k}\frac{1}{\ln(n)} \ \leq \ \frac{2}{\ln(n)} \ .$$

Let S denote the event that first cycle of the protocol succeeds. Then

$$\mathbf{P}[S] = 1 - \mathbf{P}[F] \geq 1- \ \frac{2}{\ln(n)} \ .$$

If the first cycle fails, then we can upper bound the probability that the second pass fails by the probability that the first cycle fails. This is true because the likelihood of successfully collecting all tokens at a given stage increases if some of these tokens were already collected in a previous cycle, thereby allowing for more opportunities for the uncollected tokens to be chosen. Thus, we can upper bound the expected number of cycles for the protocol by

$$\frac{1}{P[S]} \ \leq \ \frac{1}{1-\frac{2}{\ln(n)}} \ \to \ 1 \qquad \text{as } n \to \infty$$

Hence, since each cycle consists of $\log_2$n stages, each of length $n\ln(n)+n\ln(\ln(n))$, the total expected number of days till the prisoners get out is upper bounded by

$$\left(\frac{1}{1-\frac{2}{\ln(n)}}\right)(\log_2 n)\left(\ n\ln(n)+n\ln(\ln(n))\ \right) \ \to \ O(\ n\ln(n)^2\ )$$

4.Potential Better Strategy

Is there any better or vest solution to the problem? Some sort of hybrid algorithm is probably the best, to use good points of more than one strategy. Binary token strategy is certainly a good idea to start with, but something else may be better in the endgame. A hybrid given by B.Felgenhauer uses the binary token strategy to start with, but has a single counter strategy midway through. His sequence of block lengths (chosen by hand) has expected days of around 3949, and running optimization program on the variable for his strategy gives around 3890.

5. Weaker Assumption

In this section we will discuss if we can still get a solution after dropping one or more assumption made above.

5.1 Drop Assumption.1

Now the protocol must be symmetric, which means each prisoner must follow the same instruction.

Since there is only one role in whole procedure, binary token protocol works in this stronger condition, while single counter strategy doesn't work.

5.2 Drop Assumption.2

Now the initial bulb state is indeterministic.

If we still hold the assumption that prisoners can count how many days have elapsed, then as a trivial consequence of the assumption, we can have the prisoner who enters on the first day turn the bulb OFF .

5.3 Drop Assumption.3

Now prisoners have no way to count how many days have elapsed.
In this situation, binary token protocol doesn't work since it depends on prisoners' knowledge of how many days have passed. On the other hand, single counter protocol works.

5.3 Drop Assumption.2 & 3

If we have to drop both assumption, then we need to make some change to single counter strategy to solve the problem:

- If you are not the counter:
  - If the bulb is OFF, and you either have never turned the bulb ON before or have only turned the bulb once, then turn it ON.
  - If the bulb is ON, do nothing.

- If you are the counter:
  - If the bulb is OFF, do nothing.
  - If the bulb is ON, turn it OFF, and set T=T+1.
  - If T = 2n, announce that all prisoners have visited.

Notice that when T= 2n, the number can consist of all 2*n times turning ON by prisoners, or 2n*-1 times and bulb's initial condition being ON. In this way, the strategy can solve the indeterministic problem even though no one can tell if the current day is the first day.

6. Variation of the problem

The interesting riddle has many variations. Each variation assumes all the conditions in the original problem, but with some aspects altered. Among those variations, the most worth thinking one is as followings:

Now the condition for every one to be freed is that every prisoner must correctly announce (at some time). In other words: every prisoner must be sure that all prisoners have been to the room.

An amazing fact is that such protocol actually exists, just a variation of binary token strategy: each prisoner has one type of token for each prisoner who will have to announce. One cycle is given to each prisoner's attempts to collect the token destined to him, then after n of these second cycle is devoted to each prisoner, and so on. The expected run time of this strategy is in $O(\; n^2 long_2 n \;)$

However we mentioned that the binary token strategy depends on the assumption that prisoners can count how many days have elapsed. What if the assumption doesn't exist?

The good news is that solution still exists. The bad news is that the solution may have the expected run time order equal to e^n . A brief description is as following:

The light bulb is always worth one token. Any prisoner who has not announced does the following: If the lightbulb is on when he enters, then he collects the soul and turns the lightbulb off. If the lightbulb is off when he enters and he has one or more tokens, then he drops one token and turns the lightbulb on. Any prisoner who has already announced always drops any tokens that he has, and leaves any that are in the room. The average runtime can be shown by constructing an appropriate Markov chain and giving lower bounds for the chance that a given prisoner will announce in the next 200 days. Notice that when there is only one prisoner left to announce, this strategy reduces to Single Counter Strategy.

7. Conclusion

From the discussion above, we can find that each protocol is based on some important assumptions and doesn't depend on other assumption. Therefore, they have advantages on different aspect.

References

1. William Wu. 100 prisoners Light Bulb, 2002

2. P. Dehaye, D. Ford, and H. Segerman, One hundred prisoners and a lightbulb, Mathematical Intelligencer 25(4) (2003), 53–61.

3. IBM Research, 100 prisoners and a lightbulb challenge, 2002

4. Bertram Felgenhauer, 100 prisoners and a lightbulb, 2002

5. Peter Winkler , Mathematical Puzzles: A Connoisseur's Collection