

15756 Randomized Algorithms

Rohan Jain

Contents

Chapter 1

Page 2

1.1 100 Prisoners Problem

2

Chapter 1

1.1 100 Prisoners Problem

This is a problem based on a riddle:

Question

There is a jail and a warden. However, the warden is very lazy and says if they win a game, they're all set free, otherwise they all die. There is 100 boxes in a room and each box has a hidden number from 1 to 100. They are assorted in a random permutation. Each prisoner is also given a number from 1 to 100. Each prisoner is allowed to enter the room and will be allowed to open 50 boxes and if they find their number, they win. They will only live if everyone finds their number. The prisoners can't communicate with each other. What strategy should they use to maximize their chances of winning?

Naive strategy: Each prisoner opens 50 boxes at random. The probability of winning is $\frac{1}{2^{100}}$.

Theorem 1.1.1

There exists a strategy such that the probability of winning is $\geq \frac{30}{100}$.

Proof. The algorithm is as follows:

1. Each prisoner opens the box with their number.
2. If they find their number, they open the box with the number they found.
3. They continue this process until they find their number or they open 50 boxes.

Cycle notation: Consider the following arrangement, where the ordinal is the box number:

1. 7
2. 6
3. 4
4. 3
5. 8
6. 1
7. 2
8. 0
9. 5
10. 10

This can be represented as 4 individual cycles: $(1, 7, 2, 6)$, $(3, 4)$, $(5, 8, 9)$, (10) .

A critical observation to make is that prisoner i will win if and only if the cycle containing i is of length ≤ 50 . That is, everyone wins iff there are no cycles of length > 50 .

We warm up by considering opening 75 boxes instead of 50. The observation is that if anyone loses, at least 75 people will lose. Let X be a random variable representing the number of people who lose. Then, $E[X] = 25$ as each player has a $\frac{1}{4}$ chance of losing. Recall Markov's inequality:

Theorem 1.1.2

Let X be a non-negative random variable. Then, for any $t > 0$, $\Pr[X \geq t] \leq \frac{E[X]}{t}$.

Applying Markov's inequality, we have $\Pr[X \geq 75] \leq \frac{25}{75} = \frac{1}{3}$. Thus, the probability of winning is $\geq \frac{2}{3}$. Moving back to the original problem, we can apply the same logic.

$$\Pr(\text{anyone loses}) = \Pr(\text{at least one cycle of length } > 50).$$

Recall how we can count the number of cycles of length ≥ 50 . For a cycle to be exactly length $n > \frac{100}{n} = 50$, we need to choose n boxes and then permute them. Thus, the number of cycles of length n is $\frac{100!}{n \cdot (100-n)!}$. Now we have to worry about the number of arrangements of the other $100 - n$ boxes, which is $(100 - n)!$. Multiplying these two quantities, we get:

$$\frac{100!}{n \cdot (100 - n)!} \cdot (100 - n)! = \frac{100!}{n}.$$

As there are $100!$ total permutations, we realize that exactly $\frac{1}{n}$ of the permutations contain a cycle of length n . Thus, the probability of losing is:

$$\Pr(\text{anyone loses}) = \sum_{n=51}^{100} \frac{1}{n} = \sum_{n=1}^{100} \frac{1}{n} - \sum_{n=1}^{50} \frac{1}{n} = H_{100} - H_{50} \approx \ln 100 - \ln 50 = \ln 2 \approx 0.693.$$

Thus, the probability of winning is $\geq 1 - 0.693 \approx 0.307$ as desired. \odot

We move on to proving that this algorithm is actually optimal:

Theorem 1.1.3

The algorithm described above is optimal.

Proof. We consider a second version of the game, where any box that a previous prisoner has opened stays open. So if prisoner i walks in to see that i has been revealed, he just leaves. Otherwise, he opens boxes until he finds i or has opened 50 boxes. Boxes are never closed.

Lemma 1.1.1

Cycle algorithm is no better at version 2 than version 1.

Proof. Option 1: Someone from my cycle in the past lost.

Option 2: I am the first person in my cycle to enter.

In both versions, the boxes in my cycle are closed. This means that in both versions, I win iff the cycle is of length ≤ 50 .

Option 3: Someone from my cycle has already entered, and they won. This means the cycle is of length ≤ 50 , so I win in both versions. \odot

Lemma 1.1.2

All algorithms are equally good in version 2.

Proof. By symmetry: If I'm opening a box, I have to pick out of the remaining boxes, and they all have the same probability of containing my number. ☺

Together, these lemmas show that the cycle algorithm is optimal in version 1. ☺