

Guessing games

CJ Quines

April 27, 2024

Warmup

- (Classical) Dylan and Sarji are playing a game. Let $A = \{0, 1, \dots, 2^{10} - 1\}$. Dylan chooses an integer a from A , and Sarji is trying to guess it. Sarji can specify a subset $S \subseteq A$, and Dylan replies by saying whether $a \in S$. What's the least number of questions Sarji needs to find a ?
 - Write a in binary. How can Sarji learn one of a 's digits with each question?
 - Suppose Sarji only asks 9 questions. How many sequences of 9 answers can Sarji get? Show that Sarji has at least two possibilities for Dylan's number.
- (Iran TST 2022/1 [↗](#)) Morteza and Mahdi are playing a game. Morteza chooses 100 finite sets of integers, A_1, \dots, A_{100} , and Mahdi is trying to guess the elements of each set. Mahdi can say two integers $1 \leq i < j \leq 100$, and Morteza replies by giving the elements in $A_i \cap A_j$ and $A_i \cup A_j$. What's the least number of questions Mahdi needs to find the elements in each set?
 - Show that if Mahdi asks for (i, j) , (j, k) , and (k, i) , Mahdi can find A_i , A_j , and A_k . Induct to show Mahdi can win with 100 questions.
 - Suppose Mahdi only asks 99 questions. Construct the graph G with one vertex for each set, and draw an edge between two sets if Mahdi asks about them together. Show that some component of G is a tree T .
 - Suppose Morteza answers, for each question corresponding to an edge in T , that the intersection is \emptyset and the union is $\{0\}$. Show that Mahdi has at least two possibilities for the elements of each set corresponding to a vertex in T .

Thoughts

Guessing games involve a **questioner**, who asks **queries** to an **answerer**, who has a **secret**. There's mostly two kinds of problems. (Unless it's not a combinatorics problem. Then what it's about probably won't be a secret.) Anyway.

- There's problems secretly about binary:
 - We show the questioner can win with q queries by considering the binary digits of the secret, or by learning half of the secret with each query.
 - We show the questioner can't win with $q - 1$ queries, often through a counting argument. If there's more possibilities for the secret than there are possibilities for answer sequences, by pigeonhole, there's some sequence of answers that correspond to at least two secrets.
- And there's problems secretly about graphs (ish):
 - We show the questioner can win with q queries, often inductively. If the questioner wants to learn n parts of the secret, show the questioner can learn $n - 1$ parts of it.
 - We show the questioner can't win with $q - 1$ queries by converting the problem to a graph. Usually we set it up so that each part of the secret is a vertex, and each query adds an edge. Then we show there's two possible secrets that correspond to the graph.


Examples

1. (Classical) We have n lamps and n switches. Our goal is to decide which switch opens which lamp. In one move, we can specify a set of switches, toggle them, and observe which lamps are toggled. Find the least number of moves needed to learn which switch goes to each lamp.

Sketch: It's $\lceil \log_2 n \rceil$ moves. For simplicity, assume $n = 2^k$.

We show k moves are enough. Number the switches with k -digit binary numbers. For the i th move, toggle all switches whose i th digit is 1.


We show $k - 1$ moves aren't enough. The information we have for each lamp is which moves it was toggled in; there's only 2^{k-1} such sets of moves, but 2^k lamps.

2. ([Russia 2004/9/3](#) ) There are 2004 boxes, each containing a white or a black ball. It is known that a non-zero even number of balls are white. A move consists of picking two boxes and asking whether at least one of them contains a white ball. What minimal number of questions is required, after which one can indicate two boxes both containing a white ball?

Sketch: It's 4005.


We show 4005 moves are enough. Ask $(1, 2)$, $(1, 3)$, etc., if any of them is a no, then 1 is black and we're done. (Why?) Else 1 is white. Repeat for $(2, 3)$, $(2, 4)$, etc. If 2 is black we're done, else answer 1 and 2.

We show 4004 moves aren't enough. Consider the graph with vertices as balls and edges as queries. Suppose we answer u, v . If uv isn't an edge, they could both be black, contradiction. Find a vertex w such that at most one of uw and vw is an edge; it exists because $4004 < 1 + 2002 + 2002$. If uw edge, declare v and w black, u white, contradiction.

3. ([Russia 2000/9/2](#) ) Tanya chose a natural number $X \leq 100$, and Sasha is trying to guess this number. He can select two natural numbers M and N less than 100 and ask Tanya about $\gcd(X + M, N)$. Show that Sasha can determine Tanya's number with at most seven questions.

Sketch: Write X in binary, we'll find its digits right-to-left. Ask $\gcd(X + 2, 2)$. If the last digit is 1, ask $\gcd(X + 1, 4)$, otherwise ask $\gcd(X + 4, 4)$. In general, if the last k digits interpreted as binary is k' , ask $\gcd(X + 2^k - k', 2^{k+1})$.

Remark: There's lots of other solutions, but I think this is most motivated.

4. ([Kürschák 2010/1](#) ) We have n keys and n locks. Our goal is to decide which key opens which lock. In one move, we choose a key and a lock, and test whether the lock can be opened with the key. Find the least number of moves needed to learn which key goes to each lock.

Sketch: It's $\binom{n}{2}$ moves.






We show $\binom{n}{2}$ moves are enough by induction. Try the first key on every lock; this takes at most n moves. Then there are $n - 1$ keys left to match with $n - 1$ locks, by hypothesis this takes at most $\binom{n-1}{2}$ moves.

We show $\binom{n}{2} - 1$ moves aren't enough. Make a $K_{n,n}$ where one side is keys and the other side is locks; the secret corresponds to a (perfect) matching. Fail each query and remove its edge from the graph, unless we'd make a matching impossible.

After all queries we have a subgraph of $K_{n,n}$ with at least $n^2 - \left(\binom{n}{2} - 1\right)$ edges. By our answers, we know one's matching possible. Show that a cycle exists for size reasons. Its symmetric difference with the matching is another matching.


Problems

These are mostly direct applications of the above ideas.

- ([HMIC 2019/2](#) ) Annie has a permutation $(a_1, a_2, \dots, a_{2019})$ of $S = \{1, 2, \dots, 2019\}$, and Yannick wants to guess her permutation. With each guess Yannick gives Annie an n -tuple $(y_1, y_2, \dots, y_{2019})$ of integers in S , and then Annie gives the number of indices $i \in S$ such that $a_i = y_i$. Show that Yannick can always guess Annie's permutation with at most 24000 guesses.
Hint: 9
- ([TOT Spring 2016/Junior A/7](#) ) Let $n > 2$ be an integer. There are $n + 1$ good batteries and n batteries. A lamp uses two batteries, and it works only if both of them are good.
 - What is the least number of attempts needed to make the lamp work?
 - Same as (a), but with n good batteries.**Hint:** 13
- ([India 2021/4+](#) ) A Magician and a Detective play a game. The Magician lays down cards numbered from 1 to 52 face-down on a table. On each move, the Detective can point to two cards and inquire if the numbers on them are consecutive. The Magician replies truthfully. After a finite number of moves, the Detective points to two cards. She wins if the numbers on these two cards are consecutive, and loses otherwise. What's the least number of questions the Detective needs to win? **Hint:** 5
- ([Russia 2019/11/3](#) ) We are given n coins of different weights and n balances, for some $n > 2$. On each turn one can choose one balance, put one coin on the right pan and one on the left pan, and observe the result. However, one of the n balances is broken, and only shows random results. What is the least number of turns needed to find the heaviest coin? **Hints:** 7 11
- ([Russia 2008/11/6+](#) ) A magician should determine the area of a hidden convex 2008-gon $A_1A_2 \cdots A_{2008}$. In each step he chooses two points on the perimeter, where the chosen points can be vertices or points dividing selected sides in selected ratios. Then his helper divides the polygon into two parts by the line through these two points and announces the area of the smaller of the two parts. Show that the magician can find the area of the polygon in 50 steps.
Hints: 18 2

Other problems

These are less direct applications. Not necessarily harder than the last section.

- ([Brazil 2020/5](#) ) Let $n \geq k$ be positive integers. Each person in a group of n people either always tells the truth or always lies. Arnaldo can ask one of these people "In set A , are there an even number of people who always tell the truth?" where A is a subset of size k of the set of n people. The answer is always "yes" or "no".
 - For which n and k is it possible to determine which people always tell the truth, and which people always lie?
 - If it's possible, what's the minimum number of questions needed?**Hint:** 15

7. (UK 2016/2/2 [🔗](#)) Alison has compiled a list of 20 hockey teams, ordered by how good she thinks they are, but refuses to share it. Benjamin may mention three teams to her, and she will then choose either to tell him which she thinks is the weakest of the three, or which she thinks is the strongest of the three. Benjamin may do this as many times as he likes. Determine the largest N such that Benjamin can guarantee to be able to find a sequence T_1, T_2, \dots, T_N of teams with the property that he knows that Alison thinks that T_i is better than T_{i+1} for each $1 \leq i < N$. **Hints:** 4 16
8. (TOT Spring 2016/Senior A/4 [🔗](#) / IOI 2014/game [🔗](#)) There are 64 towns in a country and some pairs of towns are connected by roads but we do not know these pairs. We may choose any pair of towns and find out whether they are connected or not. Our aim is to determine whether it is possible to travel from any town to any other by a sequence of roads. Prove that there is no algorithm which enables us to do so in less than 2016 questions. **Hint:** 12
9. (Wire association [🔗](#)) Let $n > 2$ be an integer, and $S = \{1, \dots, n\}$. There's a secret permutation $\pi: S \rightarrow S$. In a move, you choose a set P , whose elements are disjoint subsets of S . You then choose $p = \pi$ or π^{-1} , and learn $\{\{p(a) \mid a \in A\} \mid A \in P\}$. (You learn a set, so the elements are unordered.) What's the smallest number of moves needed to learn π ? **Hints:** 8 14
- Remark:** Once you think you've found the answer, check your guess with this interactive version: <https://files.timwi.de/Misc/Wire%20puzzle.html>. You choose the partition by clicking some numbers and then "make group."
10. (IOI 2016/messy [🔗](#)) Let $n > 2$ be an integer, and $S = \{1, \dots, n\}$. There's a secret permutation $\pi: S \rightarrow S$. First, you choose a set P , with $|P| \leq \lceil n \log_2 n \rceil$, whose elements are subsets of S . Then you make several queries. In each query, you choose a subset A of S , and learn if there exists some $A' \in P$ such that $A = \{\pi(a) \mid a \in A'\}$. Prove that you can learn π with at most $\lceil n \log_2 n \rceil$ queries. **Hints:** 1 10
11. (Russia 2013/11/4 [🔗](#)) There are 2013 cards face-down on a table, each with a distinct number. In one move, we can point to ten cards and learn the number on one of the cards, though we don't know which. What's the largest number of cards we can guarantee to identify the numbers of? **Hints:** 3 17 6

Hints

1. Pretend you could restart from the beginning. Split the problem in half.
2. To find P , do some sort of “ternary search”.
3. There’s only one way we can be sure of a card’s number, which is if all queries that have that card as an answer intersect only in that card.
4. Draw the graph with an edge between two vertices if we don’t know who’s stronger.
5. For the lower bound, you can use Ore’s theorem.
6. If there are only 27 cards, show there’s a counterstrategy that prevents us from learning any cards, by grouping the cards into triangles.
7. Pretend one balance is correct and use it to find the heaviest coin. Do the same thing with another balance.
8. The answer is two moves.
9. Find where 1 is, then lamps-and-switches.
10. How do you make sure queries from the smaller problems don’t overlap? Add extra elements.
11. For the lower bound, show that there’s some coin that only got compared with once; what can we say about that coin?
12. Answer no unless you have to. If we knew that the graph was connected before the last edge, then those two vertices must already be joined by some path.
13. For the lower bound, you can use Turán’s theorem.
14. If you know $\pi(a) = b$ and $\{\pi^{-1}(b), \pi^{-1}(c)\} = \{\pi(a), \pi(d)\} \dots$
15. Ask everyone the same set.
16. For the upper bound, answer so that pairs are indistinguishable.
17. If there are only 18 cards, show there’s a counterstrategy that prevents us from learning any cards, by pairing the cards up.
18. Find a point P such that A_1P divides the polygon in half.

Sketches

1. Use 2018 queries to find a_1 , WLOG $a_1 = 2019$; for each subsequent query we'll fix $a_1 = 2019$. Use the lamps-and-switches method to get 1025 through 2018, each takes 11 queries. Then use lamps-and-switches to get 1 through 1024, each takes at most 10 queries.
2. For (a), answer is $n + 2$. Divide into pairs and query each. If all fail, then the remaining battery must be good, query it with both batteries of a pair. To show $n + 1$ isn't possible, need to show that a graph with $2n + 1$ vertices and $n + 1$ edges has an independent set of $n + 1$. Equivalently, its complement needs to have a clique of size $n + 1$. This follows from [Turán's theorem](#) \blacksquare . (b) is similar, but you have to do a bit more work in the construction.
3. It's 50, ask 1 with 2 through 51, if there's a yes we're done, otherwise answer 1 and 52. If ≤ 49 queries, consider graph, want to show a Hamiltonian path exists in its complement. This follows from [Ore's theorem](#) \blacksquare . (If you haven't seen it before, the statement is provable by induction; a graph such that $\deg u + \deg v \geq |V|$ for all non-adjacent u, v is Hamiltonian.)
4. It's $2n - 1$. Pick three balances A, B, C . Use A to get the heaviest coin with $n - 1$ queries, similarly use B . If they agree we're done, otherwise use C to compare. To show $2n - 2$ isn't enough, draw the graph with $u \rightarrow v$ if u weighs less than v . Some vertex v has outdegree ≤ 1 . We can't distinguish between the world where all balances were correct, and the world where v is the heaviest.
5. There's exactly one point P on the perimeter such that A_1P divides the polygon in half; say it's on side A_iA_{i+1} . Then, if M is the midpoint of A_iA_{i+1} , we can use A_1A_i , A_1M , and A_1A_{i+1} to find the area, this takes 3 queries.
 It remains to find A_iA_{i+1} . Consider the function $f: \{2, \dots, 2008\} \rightarrow \mathbb{R}$ that takes i to the answer for query A_1A_i . It's a unimodal function, and we're looking for its maximum; we use [ternary search](#) \blacksquare to find it in $\lceil 2 \log_{3/2} 2007 \rceil = 38$ queries. (It's possible to do much better, $\lceil \log_\varphi 2007 \rceil = 16$ queries, with [golden section search](#) \blacksquare .)
6. It's impossible when k is odd; we can't distinguish the worlds where everyone tells the truth or everyone lies. Otherwise we need n queries; ask everyone with the same set A .
7. It's 10. Strategy: ask every question, draw the graph with an edge between two vertices if we don't know who is stronger between them. Each vertex has degree at most 1. (If not, uv and uw are both edges, query u, v, w , this eliminates at least one edge.) Split into connected components, each one has at most two, pick a vertex in each component.
 Counterstrategy: rank the teams from 1 to 20, group them into pairs (1, 2), (3, 4), etc. If a query's teams are all from different pairs, answer whatever. If a query includes two teams from the same pair, name the third team; it's either strongest or weakest. We can't distinguish between the teams in each pair.
8. Answer no unless you have to. In other words, answer no unless it would force the graph to be disconnected. Say the algorithm made 2015 queries, and there's one edge uv left. By the way we answered, the algorithm must've concluded the graph is connected, even if uv wasn't drawn. Therefore, in the current graph, there must be some path from u to v . Let st be the last edge on that path that we answered yes to; WLOG there were paths $u \rightarrow s$, $t \rightarrow v$. We answered yes because if we answered no, the graph must be disconnected, say because there

wouldn't be a path from p to q . WLOG there's paths $p \rightarrow s, q \rightarrow t$. But then we could draw uv and find paths $p \rightarrow s \rightarrow u, v \rightarrow t \rightarrow q$.

9. Two moves. (We can't do it with one move as there are more permutations than partitions.) The idea is if we know $\pi(a) = b$ and $\pi(ac) = bd$, we can deduce $\pi(c) = d$. But we can be more efficient: if we know $\pi(a) = b$ and $\{\pi^{-1}(b), \pi^{-1}(c)\} = \{a, d\}$, we can deduce $\pi^{-1}(c) = d$. And if we know that and $\{\pi(d), \pi(e)\} = \{c, f\}$, we know $\pi(e) = f$, etc.

We do n odd first. For concreteness, we'll do an example with $n = 5$, and label the range with letters. Split S into pairs: 1, 23, 45, and query with π . Say we get the answer C, AD, BE . We split the answer into pairs the other way: CA, DB, E , and query with π^{-1} ; the above finishes. The n even case is similar: query 1, 2, 34, 56.

10. Say it takes $f(n)$ queries to solve the problem for n . Pretend for a second that we can restart from the beginning. First, let P be the singletons $\{1\}, \{2\}, \dots, \{n/2\}$. Then use n queries to find how the halves correspond. Now we have two subproblems, with sizes $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$. We have to make sure queries from subproblems don't overlap; to do this, add all elements not in the subproblem to every set in P . Recursion is $f(n) = f(\lfloor n/2 \rfloor) + f(\lceil n/2 \rceil) + n$ and $f(1) = 0$, which solves to $f(n) = n \lceil \log_2 n \rceil$.

11. It's $2013 - 27 = 1986$. We claim there's a strategy to learn one card's number when we have 28 cards; we can then induct to learn 1986 cards. Suppose we have 28 cards, ask every question. Let S_k be the intersection of all queries with answer k . If there's some $|S_k| = 1$ we're done. Otherwise, pick two vertices from each S_k and draw an edge. By construction, there's no independent sets of size 10. But our graph has 28 edges and 10 vertices, so by Turán's theorem the complement has a clique of size 10, contradiction.

Now we show a counterstrategy for 27 cards (which solves the problem). Group the vertices into nine directed triangles. By pigeonhole, each query selects two vertices in some triangle, and hence some edge $u \rightarrow v$ in that triangle; answer v . We can't distinguish between (a, b, c) or (b, c, a) for each of the triangles.