Adivinhe um número (com mentira)

Roberto Imbuzeiro Oliveira*

1 de Julho de 2019

1 Introdução

Neste texto, exploraremos a Matemática por detrás de dois jogos.

1. *Adivinhe um número:* eu penso num número entre 1 e n. Você tem que descobrir que número é esse me fazendo até q perguntas com respostas "sim"ou "não". Qual o menor número de perguntas q de que você precisa para descobrir o número com certeza?

2. *Adivinhe um número com mentira:* eu penso num número entre 1 e n. Você tem que descobrir que número é esse me fazendo até q perguntas com respostas "sim"ou "não". No entanto, eu tenho o direito (mas não a obrigação) de mentir em uma das minhas respostas. Qual o menor número de perguntas q de que você precisa para descobrir o número com certeza?

O primeiro jogo é muito conhecido. É um resultado conhecido que $q = \lceil \log_2 n \rceil$ perguntas são necessárias e suficientes para adivinhar n quando não há mentiras. Por exemplo, $q = 3$ para $n = 8$

O segundo jogo não é tão famoso, mas também é bem estudado. Ao que tudo indica, ele foi concebido independentemente por dois matemáticos: o húngaro Álfred Rényi e o polaco naturalizado americano Stanislaw Ulam. Por isso, ele às vezes é chamado de jogo de Ulam ou jogo de Rényi-Ulam. Como veremos, este jogo tem a ver com o problema de *transmissão de informação com erros*.

Provaremos que $q = 6$ perguntas são necessárias e suficientes para $n = 8$ com uma mentira. De fato, veremos um resultado um bocado mais geral.

Teorema 1. *No jogo com mentiras, se n é da forma $2^{2^\ell - 1}$ com $\ell \in \mathbb{N}\backslash\{0,1\}$, o número mínimo de perguntas que bastam é o menor $q \in \mathbb{N}$ com $n(q+1) \leq 2^q$.*

Os primeiros valores de n da foroam acima são listados, juntamente com os q para os casos sem e com mentira, na Tabela 1. Para n grande, o valor de $q = q(n)$ é

$$q(n) = \log_2 n + (1 + \epsilon(n)) \log_2 \log_2 n, \text{ com } \epsilon(n) \to 0 \text{ quando } n \to +\infty.$$

*IMPA, Rio de Janeiro, Brasil. rimfo@impa.br. Material para o PAPMEM de julho de 2019.

$n = 2^{2^\ell - 1}$	q **sem mentira**	q **com mentira**
2	1	2
8	3	6
128	7	11
32768	15	20

Tabela 1: Os valores de n da forma $2^{2^\ell - 1}$ e os números de pergunta mínimos para adivinhar o número nos casos sem mentira e com mentira. Note que o valor sem mentira é $\log_2 n = 2^\ell - 1$, como veremos abaixo. Também veremos que o q com mentira é o menor número natural com $2^q \geq (q+1)n$.

Na verdade, a conclusão do teorema acima vale para qualquer n par. Para n ímpar o resultado é um pouco diferente[1].

2 Relação com problemas de transmissão de informação

Imagine que um satélite está mandando informações para a Terra. Essas informações chegam *codificadas* em formato digital. Basicamente, isso quer dizer que cada possível mensagem que o satélite pode mandar é convertida numa sequência de 0s e 1s (ou *bits*) de acordo com uma tabela pré-definida (ou *código*). Por exemplo, a tabela pode ser:

$$\begin{aligned} \mathsf{ALIENS} &\mapsto 10010; \\ \mathsf{LOW\ BATTERY} &\mapsto 10011; \\ \mathsf{ASTEROID} &\mapsto 10110; \\ &\dots \end{aligned}$$

e assim por diante.

Podemos imaginar que um computador na Terra recebe a sequência de bits que o satélite envia. Esses bits chegam um por um e cada um deles vai revelando mais informações sobre qual palavra codificada foi enviada.

A relação com o jogo de adivinhar números é a seguinte: se imaginamos que cada bit 0 ou 1 é uma resposta "não"ou "sim", então a transmissão é como se o satélite (S) estivesse respondendo perguntas da Terra (T). Por exemplo, para a mensagem ALIENS, podemos imaginar a seguinte sequência de perguntas e respostas.

T: O primeiro bit da sua mensagem é 1?
S: Sim!

[1]Uma solução geral sobre qual é o q mínimo para um n dado é apresentada no artigo ELLIS, R. et al. "How to play the one-lie Rényi–Ulam game." *Discrete Mathematics* volume 308 (2008), Issue 23, 5805-5808. Outro artigo apresenta uma solução mais simples para o caso em que n é potência de 2: OSTHUS, D. e WATKINSON, R. "A simple solution to Ulam's liar game with one lie." *Elementary Mathematics*, Volume 63 (2008), 97-101

T: O segundo bit é 1?
S: Não!
T: O terceiro bit é 1?
S: Não!
...
T: O último bit é 0?
S: Sim!
T: Já sei! Você enviou 10010, o que quer dizer ALIENS!

Agora imagine que há *erros na transmissão dos bits*. Uma situação possível é que alguns 1s viram 0s antes de chegar a Terra ou vice-versa.

Neste caso, temos um problema. No nosso exemplo, com apenas um erro qualquer uma das sequências de bits na tabela pode virar qualquer outra. Ou seja: o código que propusemos *não é resistente a erros*.

A Teoria da Informação foi inaugurada por Claude Shannon com um artigo em 1948 [2]. Uma das suas perguntas clássicas é sobre a transmissão de informação com erros. Se admitimos certo tipo e certa taxa de erros, qual é o código mais curto que permite transmitir mensagens sem que os erros atrapalhem no resultado final? Por um lado, a presença de erros exige que haja alguma redundância no código. Por outro lado, não queremos ter redundância demais.

No nosso caso, estamos considerando o seguinte caso simples deste problema. Se o conjunto de mensagens tem n elementos, qual o tamanho mínimo q de um código que nos permite recuperar a mensagem correta *mesmo admitindo a possibilidade de que um bit chegue errado*? Este é exatamente o problema de adivinhar um número com mentira! Obviamente, na prática, também interessa o problema com mais "mentiras" ou erros.

Neste caso simples, conseguiremos construir um código ótimo diretamente. Fazer isso de forma geral é um problema bem mais difícil e envolve ideias de várias áreas de Matemática, indo desde Combinatória (Teoria de Grafos) até Álgebra (Teoria de Galois, Geometria Algébrica).

3 Resolvendo o jogo sem mentiras

Vamos pensar agora em como resolver o jogo sem mentiras, que é um bocado mais fácil que o outro.

Primeiro suporemos que $n = 2^k$ é uma potência de 2. Neste caso, a resposta é que $q = k$ perguntas são necessárias e suficientes para saber a resposta (por exemplo, três perguntas para $n = 8$). Vejamos duas maneiras de pensar na solução.

A ideia para ganhar o jogo com o número mínimo de perguntas é a seguinte: quem faz a pergunta deve *sempre tentar excluir metade das possibilidades.* Ou seja, a pergunta deve ser escolhida de modo que um "sim"exclua metade dos números que ainda restam e um "não"exclua a outra metade. Vejamos um exemplo com $n = 8$ (você faz as perguntas).

[2]SHANNON, C. E... "A Mathematical Theory of Communication". Bell System Technical Journal. 27 (3): 379–423 (1948).

(Eu penso num número entre 1 *e* 8*.)*
Você: Seu número está entre 1 e 4 (inclusive)?
Eu: Sim! *(sobraram* $1, 2, 3, 4$*)*
Você: Seu número está entre 1 e 2 (inclusive)?
Você: Não! *(sobraram* $3, 4$*)*
Eu: Seu número é o 3?
Você: Sim! *(acabou)*

Deste modo, vemos que, para $n = 2^k$, é sempre possível ganhar com k perguntas? De fato, é fácil ver que, para qualquer n, o número ótimo de perguntas é $q := \lceil \log_2 n \rceil$, que conseguimos implementar usando perguntas que dividem os números ainda possíveis em conjuntos de tamanho mais próximo possível da metade.

Por que não é possível ganhar com menos perguntas? Um argumento possível é o seguinte: cada vez que você faz uma pergunta, os números são divididos em dois grupos, correspondendo às respostas "sim"e "não". Se você der azar, o número verdadeiro está no grupo maior, que tem pelo menos metade dos elementos. Se você der azar várias vezes, vão sobrar 2^{k-1} números depois de uma pergunta, 2^{k-2} números depois de duas perguntas e assim por diante. Portanto, para que só sobre um número no final, é necessário que o número q de perguntas satisfaça $2^{k-q} \leq 1$, o que é o mesmo que pedir que $q \geq k$.

Apresentamos agora uma maneira mais formal de pensar em porque são necessárias pelo menos $k = \lceil \log_2 n \rceil$ perguntas para n qualquer. Lembre-se da relação entre o jogo e a codificação. Deste modo, a cada número natural $1 \leq x \leq n$ corresponde uma sequência $\phi(x) \in \{0, 1\}^q$ de q respostas "sim"(1) e "não"(0). Ou seja, temos uma *função*

$$\phi : \{1, \ldots, n\} \to \{0, 1\}^q$$

que faz a correspondência entre cada x e a sequência de respostas correspondente (ou ainda, à sequência de bits que corresponde a x na tabela do código).

Para que as respostas determinem x, é necessário que cada x corresponda a uma sequência *diferente* de respostas. Ou seja, é necessário que ϕ seja *injetiva*.

O *Princípio da Casa dos Pombos* nos diz que uma função injetiva $\phi : P \to C$ de um conjunto P finito (de pombos) em um conjunto C finito (de casas) com $|P| > |C|$ *não pode ser injetiva* (isto é, alguma casa tem mais de um pombo). Como nós *sabemos que* ϕ *é injetiva*, é necessário que $|P| \leq |C|$. No caso,

$$n \leq \text{número de elementos de } \{0, 1\}^q = 2^q.$$

Isso é o mesmo que pedir que $q \geq \lceil \log_2 n \rceil$.

4 O caso com mentiras: uma cota inferior geral

O raciocínio acima mostra duas coisas. A primeira é que podemos pensar nos jogos de adivinhar números tanto como problemas de *códigos* quanto como problemas de *dividir os números em dois grupos*. A segunda é que o Princípio da Casa dos Pombos é útil para mostrarmos que um certo número de perguntas é necessário.

Agora usaremos o mesmo Princípio para provar o seguinte resultado.

Teorema 2. *No jogo de adivinhar um número entre* 1 *e n com uma mentira, para que seja possível adivinhar o número com certeza, o número de perguntas deve satisfazer* $2^q \geq (q+1)n$.

Esse teorema é metade do que enunciamos na introdução. Para $n = 8$, ele mostra que é impossível adivinhar o número com 5 ou menos perguntas (confira!).

Para provar o teorema, usaremos o mesmo conjunto $C = \{0,1\}^q$ de "casas" descrito acima. Também precisamos de um conjunto de "pombos"adequado. Como encontrá-lo?

No caso sem mentira, o conjunto de pombos era $P := \{1, 2, \ldots, n\}$. O número escolhido determina a resposta a qualquer pergunta que possa ser feita. No caso atual, o que determina as respostas não é só o número, mas também *em que momento ocorre a mentira!*

Chamemos de *escolha* um par (x, i), onde $1 \leq x \leq n$ corresponde ao número escolhido e $1 \leq i \leq q+1$ correponde a dizer que o sujeito vai mentir na i-ésima resposta (ou então não vai mentir, se $i = q+1$). O conjunto de pombos que usaremos é feito destas escolhas:

$$P := \{1, \ldots, n\} \times \{1, \ldots, q+1\}.$$

Note que o par (x, i) determina completamente as respostas às perguntas feitas. Logo há uma função $\phi : P \to C$ que associa a cada $(x, i) \in P$ a sequência de respostas às perguntas.

Afirmo que esta função ϕ tem de ser injetiva. Com efeito, se, ao final das respostas, é possível dizer qual é o número x correto, também é possível verificar em que resposta i o sujeito mentiu (admitindo que "não mentir" é o mesmo que "mentir na pergunta $q+1$"). Dito de outra forma, se $(x, i) \neq (x', i')$, em algum momento as respostas têm de ser diferentes e portanto $\phi(x, i) \neq \phi(x', i')$.

Como ϕ é injetiva, o Princípio das Casas dos Pombos garante $|P| \leq |C|$, ou seja, $n(q+1) \leq 2^q$.

5 Adivinhando com uma mentira e $n = 2^{2^\ell - 1}$

Terminamos estas notas explicando como ganhar o jogo com o menor número possível de perguntas, ao menos em alguns casos.

Teorema 3. *Se* $n = 2^{2^\ell - 1}$ *com* $\ell \in \mathbb{N} \backslash \{0, 1\}$ *e* q *satisfaz* $2^q \geq n(q+1)$*, então é possível adivinhar um número entre* 1 *e n com q perguntas, mesmo permitindo uma resposta mentirosa.*

Como já provamos que não dá para ganhar com menos perguntas, isto concluirá a prova do teorema na introdução. Para constar, a estratégia ótima para o caso em que n é uma potência de 2 geral é bem parecida.

5.1 Preliminares

Para provar este teorema, há dois passos. O primeiro deles é mostrar que um q satisfazendo $n(q+1) > 2^q$ não pode funcionar, e isso já foi feito antes.

O segundo passo é provar que, se $2^q \geq n(q+1)$, há uma estratégia que sempre funciona.vamos usar uma estratégia que sempre funciona para n tem a forma estipulada no teorema. Para facilitar, escrevemos $k := 2^\ell - 1$, de modo que $n = 2^k$

Para descrever nossa estratégia, precisamos acompanhar o que acontece com os números de 1 a n ao longo das perguntas e respostas. Em cada momento, falaremos de três conjuntos de números: os *saudáveis*, os *feridos* e os *mortos*.

- Um número entre 1 e n é dito *saudável* se é compatível com todas as respostas dadas até o momento.
- Um número é dito *ferido* se é compatível com todas as respostas, menos uma.
- Um número é dito *morto* se é incompatível com duas ou mais respostas.

Chamaremos de x o número a ser adivinhado. Em qualquer momento, x pode estar saudável ou ferido, mas nunca pode estar morto. Afinal, um número morto foi "negado" duas ou mais vezes, mas, como apenas uma mentira é permitida, o número correto só pode ser negado uma vez. Por outro lado, um número saudável ou ferido é um número que não foi excluído até aquele momento.

Ao longo das rodadas, nós acompanharemo que chamamos de *estado* do jogo. O estado é um par (s, f), onde s é a quantidade de números de 1 a n saudáveis e f é a quantidade de feridos. Nosso objetivo é chegar ao final do jogo com $s + f = 1$; isto é, apenas um número não deve estar morto no fim.

Nossa estratégia para vencer o jogo é bem simples.

> **Estratégia:** *Na j-ésima rodada, escolheremos um conjunto B_j que contem:*
>
> - *metade dos números saudáveis, arredondando* para cima *se necessário;*
> - *metade dos números feridos, arredondando* para baixo *se necessário.*
>
> *A pergunta feita é: O número x está em B_j? Só se deve parar de fazer perguntas quando só sobrar um número vivo.*

Um exemplo do jogo progredindo é dado na Tabela 2.

De acordo com a resposta, a quantidade de números saudáveis e feridos vai se alterar. Veremos nas próximas subseções que, depois de r ou menos perguntas, chegaremos ao estado $(1, 0)$ ou ao estado $(0, 1)$.

5.2 Etapa 1: as primeiras $k = 2^\ell - 1$ perguntas.

Num primeiro momento, o estado é $(2^k, 0)$: todos os números são saudáveis e nenhum está ferido. O conjunto B_1 conterá exatamente metade dos níumeros saudáveis. Por isso, o próximo estado necessariamente será $(2^{k-1}, 2^{k-1})$, não importanto a resposta.

Na *segunda pergunta*, escolhe-se um subconjunto B_2 contendo *exatamente* metade dos números saudáveis e metade dos feridos. Seja lá qual for a resposta, metade dos saudáveis ficará ferida e metade dos feridos morrerá. Portanto, o novo estado do jogo é $(2^{k-2}, 2^{k-2} + 2^{k-2}) = (2^{k-2}, 2^{k-1})$.

Imagine que antes da i-ésima pergunta temos s_{i-1} números saudáveis e f_{i-1} feridos. Suponha ainda que ambos os números são pares (veremos que isso é verdade depois). A *i-ésima pergunta* será feita como a segunda, selecionando um conjunto B_i contendo

Rodada	Saudáveis	Feridos	Está em ... ?	Resposta
1	$\{1,2,3,4,5,6,7,8\}$	$\emptyset$	$\{1,2,3,4\}$	SIM
2	$\{1,2,3,4\}$	$\{5,6,7,8\}$	$\{3,4,5,6\}$	NÃO
3	$\{1,2\}$	$\{3,4,7,8\}$	$\{1,3,4\}$	SIM
4	$\{1\}$	$\{2,3,4\}$	$\{1,2\}$	NÃO
5	$\emptyset$	$\{1,3,4\}$	$\{1,3\}$	SIM
6	$\emptyset$	$\{1,3\}$	$\{1\}$	NÃO
Final	$\emptyset$	$\{\mathbf{3}\}$	-	-

Tabela 2: A tabela acima relata um jogo em que $x = 3$ e $n = 8$. Nas linhas de 1 a 6, temos os conjuntos de números saudáveis e feridos antes de ser feita a pergunta daquela rodada, que é representada na segunda coluna mais à direita. A última linha mostra os conjuntos finais de saudáveis e feridos: note que, se tudo der certo, um destes é vazio e o outro tem um elemento. O conjunto B usado na pergunta aparece na quarta coluna e a resposta está na coluna mais à direita. Note que, de uma linha para outra, cada número saudável incompatível com a resposta fica ferido e cada número ferido incompatível morre ("some").

metade dos saudáveis e metade dos feridos e perguntando: o número misterioso está em B_i? É fácil ver que, novamente, metade dos saudáveis vira ferido e metade dos feridos morre. Portanto, o novo estado do jogo é :

$$(s_i, f_i) = \left(\frac{s_{i-1}}{2}, \frac{f_{i-1}}{2} + \frac{s_{i-1}}{2}\right).$$

Qual é o estado do jogo depois de i perguntas? Vamos testar o estado inicial e os obtidos depois das quatro primeiras perguntas:

$$(2^k, 0) \mapsto (2^{k-1}, 2^{k-1}) \mapsto (2^{k-2}, 2^{k-1}) \mapsto (2^{k-3}, 3.2^{k-3}) \mapsto (2^{k-4}, 4.2^{k-4}).$$

Agora já dá para adivinhar uma fórmula: por indução, podemos provar que, depois de $0 \leq i \leq k$ perguntas, o estado é

$$(s_i, f_i) = (2^{k-i}, i.2^{k-i}).$$

Em particular, para $i < k$ tanto s_i quanto f_i são pares e nossa estratégia funciona. Assim, o estado ao fim da Etapa 1 é $(1, k) = (1, 2^\ell - 1)$. Ou seja, ao fim da primeira etapa, sobram exatamente $k = 2^\ell - 1$ números feridos e um saudável; os outros todos morreram e podem ser desconsiderados.

5.3 Etapa 2: as últimas $q - 2^\ell + 1$ perguntas

Na Etapa 1, foram feitas q perguntas e chegamos ao estado $(1, 2^\ell - 1)$. Restam

$$r := q - k = q - 2^\ell + 1 \text{ perguntas a serem feitas.}$$

Veja que, como $2^q \geq (q+1)n = (q+1)2^k$, temos:

$$2^r \geq (q+1) \text{ e também } q \geq k+1,$$

portanto

$$2^r \geq k+2 = 2^\ell + 1, \text{ ou seja, } r \geq \ell + 1.$$

Continuaremos a seguir nossa estratégia no restante do jogo. A seguinte observação será importante.

Observação 1. *Se o jogo está no estado $(1, 2^t - 1)$, com $t \geq 1$ natural, e uma pergunta é feita de acordo com a nossa estratégia, então o novo estado é $(1, 2^{t-1} - 1)$ ou $(0, 2^{t-1} + 1)$.*

Prova: Suponha que estamos na j-ésima pergunta. Como só temos 1 elemento saudável, ele deverá estar no conjunto B_j correspondente. Além disso, o conjunto de feridos tem número ímpar de elementos, logo B_j deverá conter a metade desses números arredondando para baixo: ou seja, $2^{t-1} - 1$ elementos. Os outros 2^{t-1} números feridos ficam de fora de B_j.

A pergunta é: o número x pertence a B_j?

- Se a resposta é *sim*, o novo estado do jogo é $(1, 2^{t-1} - 1)$: afinal, todos os 2^{t-1} números feridos fora de B_j são mortos.
- Se a resposta é *não*, o número que estava saudável é ferido e morrem os $2^{t-1} - 1$ números feridos que estavam em B_j. Portanto, sobram $2^{t-1} + 1$ números feridos e nenhum saudável e o novo estado é $(0, 2^{t-1} + 1)$.

□

Agora considere que fazemos perguntas do tipo acima repetidamente a partir do estado $(1, 2^\ell - 1)$ que temos no fim da Etapa 1. Das duas, uma:

- Se obtemos a sequência de estados $(1, 2^{\ell-i} - 1)$ para $i = 1, 2, \ldots, \ell$, chegamos a $(1, 0)$ com $\ell \leq r - 1$ perguntas: agora só há um número que não está morto e este tem de ser o x correto;
- Por outro lado, pode ser que cheguemos $(0, 2^{\ell-i} + 1)$ depois de fazermos $i \leq \ell$ perguntas.

O último passo é mostrar que adivinharemos o número no segundo caso, chegando ao estado $(0, 1)$. O ponto crucial é que, *como $r \geq \ell + 1$, ainda restam $r - i \geq \ell + 1 - i$ perguntas*. Portanto 2^{r-i} é maior ou igual à quantidade de números feridos:

$$2^{r-i} \geq 2^{\ell+1-i} \geq 2^{\ell-i} + 1, \text{ ou } r - i \geq \lceil \log_2(2^{\ell-i} + 1) \rceil.$$

Como só restaram números feridos, a partir de agora, qualquer número inconsistente com uma resposta será morto. O tipo de pergunta que nossa estratégia nos pede para fazer sempre matará no mínimo a metade (arredondada para baixo) dos números que ainda estão vivos. Estamos de novo na situação do jogo sem mentiras!

Mais concretamente, podemos verificar que, com mais $\ell - i$ perguntas, passaremos pelos estados:

$$(0, 2^{\ell-i-1} + 1), (0, 2^{\ell-i-1} + 1), \ldots, (0, 2^{\ell-\ell} + 1) = (0, 2).$$

Ainda resta uma pergunta (pois $r > \ell$) e ela nos permite decidir qual dos dois números restantes é o correto. Portanto, as $r - i$ perguntas são suficientes para ficarmos com um único candidato a x no final.