GT Lecture 10

Date / /

Unreliable Tests

The case of one lie (Group testing model)

Queries : Is a set containing the positive item ?

(Ulam's game : Is the item in the interval $(x, n]$?)

e.g. Find a number from $[1, 64]$. (Say, 45)

Queries of Ulam's game are : (No lies!)

$x > 32$, Yes

$x > 48$, No

$x > 40$, Yes

$x > 44$, Yes

$x > 46$, No

$x > 45$, No

$\Rightarrow x = 45$.

(•) There is no difference between this idea and a group testing model in a reliable test.

(••) By using queries of Ulam's game, Rivest et al. and Spencer are able to answer Ulam's one lie game : $M'(1, 10^6) = 25$ or 26.

Date / /

No. 2

J. Spencer, Guess a number-with lying, Math. Mag. 57, No. 2 (1984), 105-108.

(***) Using GT model, A. Pelc proves that $M'(1, 10^6) = 25$.

(*) A. Pelc, Solution of Ulam's problem on searching with a lie, J. Combin. Th. (A) 44, 129-140 (1987).

His idea is as follows.

Definition

Truth-set A : the set of items (elements) satisfying all positive outcomes.

Lie-set B : the set of items satisfying all positive outcomes except one. Here, in the case of one lie.

$|A| = a$, $|B| = b$.

- (a, b) is called a state of the game.
- The weight of a state (a,b) corresponding to a stage of the game at which "j" queries remain to be asked is

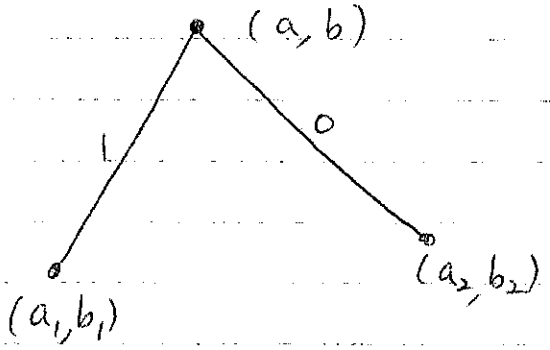
$$w_j(a,b) = a(j+1) + b.$$ → No more lie in the lie set!

$(j+1)$: each item in the truth set gives $j+1$ possibilities (number) of lying to one of the remaining j queries or not lying at all.

- Any query asked in the state (a,b) yields two states (a_1,b_1) and (a_2,b_2) corresponding to the answers Yes or No "1" "0" respectively.

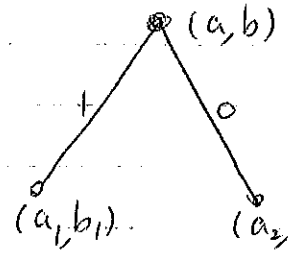
(*) Assume that the query asked is:

Is S containing the positive item (number) where $|S \cap A| = x$ and $|S \cap B| = y$? (A is the truth set and B is the lie set.)



(**) $(a_1,b_1) = (x, a-x+y)$, $(a_2,b_2) = (a-x, b-y+x)$.

- x: Truth set 變小
- $a-x+y$: 剩下的部份已不在 Truth set 所以移到 Lie set, y 則保留
- $a-x$: x 个 item 從 Truth set 中移除
- $b-y+x$: y 个 items 移除(有兩次"0"), 而 x 則換到 Lie set



$w_j(a,b)$

$w_{j-1}(a_1,b_1)$, $w_{j-1}(a_2,b_2)$

(***) $w_{j-1}(a_1,b_1) + w_{j-1}(a_2,b_2)$

$= (j-1+1)a_1 + b_1 + (j-1+1)a_2 + b_2$

$= jx + a - x + y + j(a-x) + b - y + x$

$= aj + a + b = a(j+1) + b = w_j(a,b)$.

Fact 1 The total weights remain unchanged!

Fact 2

(1) If n is even and $n(k+1) > 2^k$, then $M'(1,n) > k$.

(2) If n is odd and $n(k+1) + (k-1) \geq 2^k$, then $M'(1,n) > k$.

Proof. (1) Assume that α is an algorithm which can identify the positive item in k tests. Then, in the computation tree, the leaves are of states either $(0,1)$ or $(1,0)$. Their weights are 1 or $j+1$ (if j queries are needed later). This implies that the length of the path from the root to the leaf is $k-j$ $(k > j \geq 0)$.

Therefore, the weight of the leaf is at least $\frac{n(k+1)}{2^{k-j}}$ (j more tests (queries) are needed.) Now, since $n(k+1) > 2^k$, the weight of the leaf is larger than 2^j, i.e., $1+j > 2^j$ or $1 > 2^j$. Both of them are not possible. Hence, $M'(1,n) > k$.

(2) Since $w_k(n, 0) = n(k+1) = w_{k-1}(a_1, b_1) + w_{k-1}(a_2, b_2)$,

$\max(w_{k-1}(a_1, b_1), w_{k-1}(a_2, b_2)) \geq \frac{n+1}{2}k + \frac{n-1}{2}$.

↑ either $n-x$ or $x \geq \frac{n+1}{2}$

By assumption, $\frac{n+1}{2}k + \frac{n-1}{2} > 2^{k-1}$. This implies that of (2) after $k-1$ further tests, there is a state of weight at least 2. Furthermore, this state is not $(0,1)$ (?), thus we have the proof. ∎

F - $\forall k$ $n(k+1) > 2^k$

Fact 3 $M'(1,n) \geq \min\{k \mid n(k+1) \leq 2^k\}$, n is even.

$M'(1,n) \geq \min\{k \mid n(k+1)+(k-1) \leq 2^k\}$, n is odd.

For example, $n = 11$, $k = 7$. $n = 12$, $k = 7$.

Definition (Character of a state)

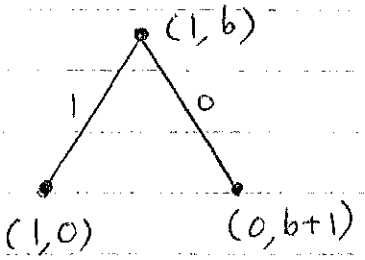
The character of a state (a,b) is defined as

$$ch(a,b) = \min_h \{ w_h(a,b) \le 2^h \} = \min_h \{ a(h+1)+b \le 2^h \}.$$

e.g. $ch(1,5) = 4$. $1\cdot(h+1)+5 \le 2^h \Rightarrow h \ge 4$

Lemma For $b \in \mathbb{N}$ and $k = ch(1,b)$, there exists an algorithm identifying the positive, starting from the state $(1,b)$ in k more tests.

Proof. By induction on b. Clearly, if $b=1$, then $ch(1,1)$ $= \min\{h \mid h+1+1 \le 2^h\} = 2$. Two more tests are needed to find the positive (?). By the following figure, if $b < k$, then



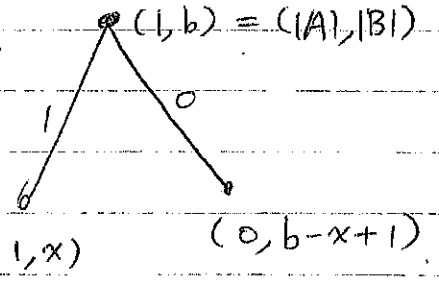
$$w_{k-1}(0, b+1) = b+1 \le k \le 2^k.$$

This implies that k more tests will be enough. For $(1,0)$ state, it is done.

(•) If the truth set is ϕ, then no more lies in the following tests.

On the other hand, let $b \geq k$.

Let $x = \left\lfloor \frac{b-k+1}{2} \right\rfloor$. Let S be a set satisfying $A \subseteq S$ and $|S \cap B| = x$. By the state diagram, we know that



$w_{k-1}(1, x) + w_{k-1}(0, b-x+1)$

$= k + x + b - x + 1$

$= k + 1 + b = w_k(1, b)$. Since $k = ch(1, b)$, $w_k(1, b) \leq 2^k$.

But, $|w_{k-1}(1, x) - w_{k-1}(0, b-x+1)|$

$= |(k+x) - (b-x+1)| = |(k-b-1) + 2x| = |2x - (b-k+1)| \leq 1$

by the choice of x.

Hence $w_{k-1}(1, x) \leq 2^{k-1}$ and $w_{k-1}(0, b-x+1) \leq 2^{k-1}$ and thus $ch(1, x) \leq k-1$ and $ch(0, b+1-x) \leq k-1$. By induction, $k-1$ more tests are needed. This concludes the proof. ∎

(*) The choice of x provides an idea to carry out the algorithm with as less tests as possible. For details please refer to the paper attached.

Date / / No. 8

(*) $|x-y| \le 1$ and $x+y \le 2^k$, let $x \le y$.

$\Rightarrow y - x \le 1$ and $x+y \le 2^k$

$\Rightarrow x \ge y-1$. thus $2y-1 \le 2^k$, $2y \le 2^k+1$.

If $y \ge 2^{k-1}$, then $2y \ge 2^k+2$. Hence $y \le 2^{k-1}$, so is x.

Lemma Let (a,b) be a state such that $b \ge a-1 \ge 1$. Then there exists a test in this state yielding states (a_1, b_1) and (a_2, b_2) such that

(1) $\left\lfloor \frac{a}{2} \right\rfloor \le a_1 \le \left\lfloor \frac{a+1}{2} \right\rfloor$, $\left\lfloor \frac{a}{2} \right\rfloor \le a_2 \le \left\lfloor \frac{a+1}{2} \right\rfloor$;

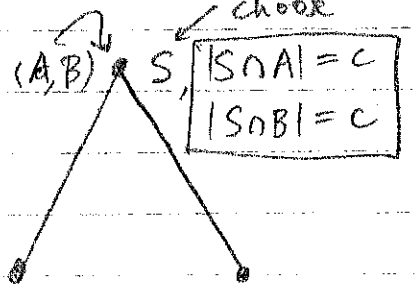
(2) $b_1 \ge a_1 - 1$, $b_2 \ge a_2 - 1$; and

(3) $ch(a_1,b_1)$, $ch(a_2,b_2) \le ch(a,b) - 1$.

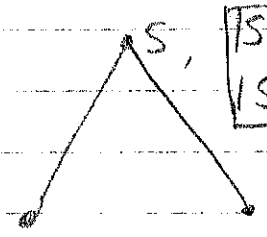
Proof.

Case 1. $b = a-1 \ge 1$

a is even, $a = 2c$



a is odd, $a = 2c+1$

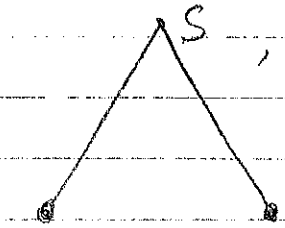


$l = ch(a, a-1) \le c$

Date / /

No. 9

Case 2 $b > a-1 \geq 1$



. . .

The idea of handling more lies, say "three".

Then, each state can be represented by a 4-tuple (a,b,c,d)

where $|A| = a$, $|B| = b$, $|C| = c$ and $|D| = d$.

A: $e \in A$ iff none of the answers is a lie (Truth set)

B: $e \in B$ iff exactly one of the answers is a lie (One lie set)

C: $e \in C$ iff exactly two of the answers are lies (Two lies set)

D: $e \in D$ iff exactly three of the answers are lies (Three lies set)

positive item (real)

Queries: "Is $e \in Q$?" where $Q = X \cup Y \cup Z \cup V \subseteq N$,

and $X \subseteq A$, $Y \subseteq B$, $Z \subseteq C$ and $V \subseteq D$.

Date / /

No. 10

(A,B,C,D) ← state of sets

$Q = X \cup Y \cup Z \cup V$

(Is $e \in Q$?)

↑ positive item

1 0

$(A \cap Q,\ (A \cap \bar{Q}) \cup (B \cap Q),\ (B \cap \bar{Q}) \cup (C \cap Q),\ (C \cap \bar{Q}) \cup (D \cap Q))$

$(A \cap \bar{Q},\ (A \cap Q) \cup (B \cap \bar{Q}),\ (B \cap Q) \cup (C \cap \bar{Q}),\ (C \cap Q) \cup (D \cap \bar{Q}))$

(a,b,c,d)

$|X| = x,\ |Y| = y,\ |Z| = z,$
$|V| = v$
$X \subseteq A,\ Y \subseteq B,\ Z \subseteq C,\ V \subseteq D.$

1 0

$(x,\ a-x+y,\ b-y+z,\ c-z+v)$

$(a-x,\ x+b-y,\ y+c-z,\ z+d-v)$

Weight of (a,b,c,d) : with k more tests to do.

$$\left(a \cdot \binom{k}{3} + b \cdot \binom{k}{2} + c\binom{k}{1} + d\right) = w_k(a,b,c,d).$$

In truth set, there are three lies to come!

In lie sets, there are 2, 1, 0 lies to come!

Definition

The character of the state (a,b,c,d),

$$ch(a,b,c,d) = \min\{ k \mid W_k(a,b,c,d) \le 2^k \}$$

Similar but more complicate arguments are needed in obtaining the answer.

So, find $M^3(4,10^6)$.

Please refer to:

A. Negro and Sereno,

1. Solution of Ulam's problem on binary search with three lies, JCT(A) 59 (1992), 149-154.

2. An Ulam's searching game with three lies, Advances in Applied Math., Vol. 13, 4, 1992, 404-428.

Two lies

W. Guzicki, Ulam's searching game with two lies, JCT(A) 54, 1990.

JOURNAL OF COMBINATORIAL THEORY, Series A 44, 129–140 (1987)

Solution of Ulam's Problem on Searching with a Lie

ANDRZEJ PELC

Département d'Informatique, Université du Québec à Hull, Case postale 1250, Succursale "B", Hull, Québec J8X 3X7, Canada

Communicated by the Managing Editors

Received March 28, 1986

S. M. Ulam, ("Adventures of a Mathematician," Scribner's, 1976.) stated the following problem: what is the minimal number of yes–no queries needed to find an integer between one and one million, if one lie is allowed among the answers. In Rivest *et al.* (*J. Comput. System Sci* 20, 396–404 (1980) and Spencer, (*Math. Mag.* 57, 105–108 (1984) partial solutions were given by establishing bounds for the minimal number of queries necessary to find a number in the set $\{1,\ldots, n\}$. Applied to the original question both solutions yield two possibilities: 25 or 26. We give an exact solution of Ulam's problem in the general case. For $n = 10^6$ the answer turns out to be 25. We also give an algorithm to perform the search using the minimal number of queries. © 1987 Academic Press, Inc.

1. INTRODUCTION AND TERMINOLOGY

The problem of coping with erroneous information in search procedures was recently studied by several authors (cf. [3, 2, 4]). The problem has a continuous version—searching for a number in a given interval, and a discrete version—searching for a number in $\{1,\ldots, n\}$. It is assumed that some of the answers to yes–no queries stated while searching may be erroneous. Obviously one has to restrict the possibilities of lying, otherwise no information can be obtained. One way of doing it is to restrict the number of possible lies.

In the continuous version it is clearly impossible to determine a number exactly in finitely many questions, even without lies. In this context the task is to minimize the measure of a set containing the unknown element. In the discrete version however, it becomes possible to find the hidden element and it seems interesting to know the minimal number of queries necessary to do so if a restricted number of lies is allowed.

The first to state this problem was Ulam [5]. He asked what is the minimal number of yes–no queries needed to find an integer between one and one million, if one lie is allowed.

0097-3165/87 $3.00

Some partial answers to this question are known. The results of Rivest *et al.* [3] applied to the case of searching for a number in $\{1,\ldots,n\}$ with at most one lie allowed give the following:

— it is impossible to determine the number in k questions if $2^k < n(k+1)$;

— it is possible to determine the number in k questions if $2^{k-1} \geq nk$.

The analysis performed by Spencer [4] provides a different bound in this context:

— it is impossible to determine the number in k questions if $2^k < n(k+1)$;

— it is possible to determine the number in k questions if $2^k \geq \frac{8}{5}n(k+1)$.

In none of the cases do the bounds determine exactly the minimal required number of questions. Applied to the original problem ($n = 10^6$) they both leave two possibilities for it: 25 or 26. Spencer [4] concludes his paper with the comment that "it seems very difficult to determine whether the answer to Ulam's original problem is twenty-five or twenty-six".

The aim of the present paper is to provided an exact answer to Ulam's question. We actually prove a general result which determines precisely the minimal number of yes–no queries necessary to find an integer in the set $\{1,\ldots,n\}$ if at most one lie is allowed. We also give an algorithm to perform the search using this minimal number of questions.

The main difference between our approach and that of Rivest *et al.* [3] and Spencer [4] is that in those two papers only comparison queries (i.e., of the form $x < a$?) were considered whereas we deal with general yes–no queries (i.e., of the form $x \in T$? for any subset $T \subset \{1,\ldots,n\}$), following the formulation of Ulam's original question. We do not know if the exact answer is the same when only comparison queries are allowed, the least we can say however is that this additional restriction would make the algorithm more complicated, due to smaller possibilities of balancing weights. This should be compared to the following comment from [3]: "It is worth noting that in none of the cases we consider does the restriction to comparison questions cause any significant increase in the number of questions needed in the worst case."

We shall consider the task of searching in terms of a game between two players, the Questioner and the Responder. This approach was adopted in [2, 4]. We follow some of the terminology from [4]. If a number x has to be found in the set $\{1,\ldots,n\}$ after k queries, the corresponding game is called the $[n, k]$ game. The rules are the following: the Responder thinks of a number x between 1 and n. Then each move of the Questioner consists of a query $x \in T$?, for $T \subset \{1,\ldots,n\}$ and the next move of the Responder is the answer yes or no. He can lie at most once. We say that the Questioner wins the $[n, k]$ game if he has a winning strategy to determine provably the

number x in k queries, i.e. a method which enables him to find x after k questions independently of the Responder's answers and to prove that x has to be the number he determined, if the opponent played according to the rules. We say that the Responder wins the $[n, k]$ game if he has a "Devil's Strategy" to win it. By this we mean that he needs not actually think of any number at the beginning but just reply "almost consistently", i.e. in such a way that at any stage of the game there is a number $x \in \{1,..., n\}$ satisfying all of his answers possibly except one. It is clear that the Questioner wins if and only if he has a winning strategy against the Responder's Devil's Strategy. Hence this definition of the Responder's win has the advantage of making the game determined: the Questioner wins the game if and only if the Responder does not. It eliminates the possibility of the Responder's loss "by chance", in spite of the lack of Questioner's winning strategy and because of his gambling luck.

With each stage of the game, when the turn of the Questioner comes, we associate a *state* of the game which is a couple (a, b) of natural numbers. The first number is the size of the truth-set: the set of those elements of $\{1,..., n\}$ which satisfy all answers given previously. The second number is the size of the lie-set: the set of those elements of $\{1,..., n\}$ which satisfy all but one answer.

Following the idea of Berlekamp [1] we define the weight of a state (a, b) corresponding to a stage of the game at which j questions remain to be asked. This weight is defined by

$$w_j(a, b) = a(j+1) + b.$$

Following Spencer [4] this can be interpreted by the fact that each number in the truth-set gives $j+1$ possibilities of lying to each of the remaining j questions or not lying at all. In the lie-set the Responder is forced to say the truth till the end, so each number in this set yields just this one possibility.

Any question asked in the state (a, b) yields two states (a_1, b_1) and (a_2, b_2) corresponding to the answers yes and no respectively. The question "Is the unknown number in the subset A of size x of the truth-set or in the subset B of size y of the lie-set?" will be referred to as the question about x elements of the truth-set and y elements of the lie-set. The states (a_1, b_1) and (a_2, b_2) yielded by a question will be always written in the order: first the one corresponding to answer yes.

It is easy to see that if some question in state (a, b) yields states (a_1, b_1) and (a_2, b_2) then

$$w_k(a, b) = w_{k-1}(a_1, b_1) + w_{k-1}(a_2, b_2).$$

For any state (a, b) we define its character as the number $ch(a, b) = \min\{k: w_k(a, b) \leqslant 2^k\}$.

2. The Main Result

The present section is devoted to the formulation and proof of the main result of this paper: for any natural n we give a necessary and sufficient condition on k for which the Questioner wins the $[n, k]$ game. The least such k for a given n is the solution of Ulam's problem, in particular for n equal to one million this k turns out to be 25.

The first lemma establishes conditions on k (for any given n), under which the Responder wins the $[n, k]$ game using the Devil's Strategy. Part (a) follows from the above-mentioned results of Rivest [3] and Spencer [4], but we include the short proof to make the paper self-contained.

Lemma 1. (a) *For even n the Responder wins the $[n, k]$ game if $n(k+1) > 2^k$.*

(b) *For odd n the Responder wins the $[n, k]$ game if $n(k+1) + (k-1) > 2^k$.*

Proof. (a) We let the Responder play the Devil's Strategy consisting in always choosing the state of non-smaller weight out of the two states yielded by the query. Since the weight of the initial state is $n(k+1) > 2^k$, after $\leqslant k$ questions the weight of the resulting state will be at least 2. All we have to do is prove that such a state cannot be equal to $(1, 0)$. (This would be the only possibility of the Questioner's win in a state of weight bigger than 1). However this could occur only if the previous state was $(1, c)$, t questions remained and the question about the unique element of the truth-set yielded states $(1, 0)$ and $(0, c+1)$ with

$$w_{t-1}(1, 0) \geqslant w_{t-1}(0, c+1)$$

indicating that the Responder answers yes. This means

$$t \geqslant c+1.$$

Since the state $(1, c)$ was reached after $k-t$ questions following the described strategy, we must have

$$w_t(1, c) = t + c + 1 > 2^k \cdot 2^{-(k-t)} = 2^t.$$

Hence by the previous inequality

$$t > 2^{t-1}$$

which is always false.

(b) For odd n, any question asked at the beginning of the $[n, k]$ game yields states (a_1, b_1) and (a_2, b_2) such that

$$\max(w_{k-1}(a_1, b_1), w_{k-1}(a_2, b_2)) \geqslant \frac{n+1}{2}k + \frac{n-1}{2}.$$

Hence the described Devil's Strategy yields a state of weight at least $((n+1)/2)k + (n-1)/2$ after the first question. If

$$n(k+1) + (k-1) > 2^k \qquad \text{then} \qquad \frac{n+1}{2}k + \frac{n-1}{2} > 2^{k-1},$$

hence after $(k-1)$ further questions this strategy yields a state of weight at least 2. It can be shown as above that this state is not $(1, 0)$ and hence it yields the Responder's win.

We next analyze the Questioner's winning possibilities.

LEMMA 2. *Let n be a natural number and $k = \mathrm{ch}(1, n)$. The Questioner wins in at most k questions starting from the state $(1, n)$.*

Proof. We prove the lemma by induction on n. Suppose that for $m < n$ it is true.

If $n < k$ then the question about the unique element of the truth-set yields states $(1, 0)$ and $(0, n+1)$ with

$$w_{k-1}(1, 0) \geqslant w_{k-1}(0, n+1).$$

The state $(1, 0)$ is already the Questioner's win and

$$w_{k-1}(0, n+1) \leqslant 2^{k-1},$$

hence the Questioner wins in at most $k-1$ questions starting from $(0, n+1)$ (since the truth-set is empty, the Responder cannot lie anymore). It follows that in the case $n < k$ the Questioner wins in at most k questions starting from $(1, n)$.

If

$$n \geqslant k \qquad \text{then let} \qquad x = \left\lceil \frac{n-k+1}{2} \right\rceil.$$

The question about the unique element of the truth-set and x elements of the lie-set yields states $(1, x)$ and $(0, n+1-x)$. We have

$$\begin{aligned} |w_{k-1}(1, x) - w_{k-1}(0, n+1-x)| &= |(k+x) - (n+1-x)| \\ &= |k-n-1+2x| \leqslant 1, \end{aligned}$$

which implies $\mathrm{ch}(1, x), \mathrm{ch}(0, n+1-x) \leqslant k-1$.

The Questioner wins in at most $k-1$ questions starting from $(0, n+1-x)$ and, by the inductive hypothesis, also starting from the state $(1, x)$. Hence, if $n \geqslant k$, he wins in at most k questions starting from $(1, n)$, which finishes the proof.

LEMMA 3. *Let (a, b) be a state such that $b \geqslant a - 1 \geqslant 1$. Then there exists a question in this state yielding states (a_1, b_1) and (a_2, b_2) such that:*

1. $\left\lceil \frac{a}{2} \right\rceil \leqslant a_1 \leqslant \left\lceil \frac{a+1}{2} \right\rceil, \left\lceil \frac{a}{2} \right\rceil \leqslant a_2 \leqslant \left\lceil \frac{a+1}{2} \right\rceil$;
2. $b_1 \geqslant a_1 - 1, b_2 \geqslant a_2 - 1$;
3. $\mathrm{ch}(a_1, b_1), \mathrm{ch}(a_2, b_2) \leqslant \mathrm{ch}(a, b) - 1$.

Proof. We first prove the lemma for $b = a - 1 \geqslant 1$. Let $l = \mathrm{ch}(a, b)$ and consider two cases.

1. a is even. Let $a = 2c$, then $b = 2c - 1$. The question about c elements from the truth-set and c elements from the lie-set yields states $(c, 2c)$ and $(c, 2c-1)$. Conditions 1 and 2 are clearly satisfied. For Condition 3 it suffices to show

$$w_{l-1}(c, 2c) \leqslant 2^{l-1}.$$

Indeed, since

$$w_l(2c, 2c-1) = 2c(l+1) + 2c - 1 = 2c(l+2) - 1 \leqslant 2^l$$

and the number $2c(l+2)-1$ is odd, we get

$$2c(l+2) \leqslant 2^l,$$

hence

$$w_{l-1}(c, 2c) = cl + 2c = c(l+2) \leqslant 2^{l-1}.$$

2. a is odd. Let $a = 2c+1$, then $b = 2c$. We first assume $a > 5$. The question about $c+1$ elements from the truth-set and $c - [\frac{l}{2}]$ elements from the lie-set yields states $(c+1, 2c - [\frac{l}{2}])$ and $(c, 2c + [\frac{l}{2}] + 1)$. Condition 1 is satisfied. In order to prove Condition 2 it suffices to show that $2c - [\frac{l}{2}] \geqslant c$ which means $[\frac{l}{2}] \leqslant c$. Indeed, since $a \geqslant 6$ we get

$$a(a+1) + (a-1) \leqslant 2^a,$$

hence

$$l = \mathrm{ch}(a, a-1) \leqslant a$$

which implies $[\frac{l}{2}] \leqslant c$.

We finally prove Condition 3

$$w_{l-1}\left(c+1, 2c-\left[\frac{l}{2}\right]\right)=(c+1)\,l+2c-\left[\frac{l}{2}\right]=c(l+2)+l-\left[\frac{l}{2}\right]$$

and

$$w_{l-1}\left(c, 2c+\left[\frac{l}{2}\right]+1\right)=c(l+2)+\left[\frac{l}{2}\right]+1.$$

If l is even then $w_l(2c+1, 2c)=2c(l+2)+l+1$ is odd and hence $w_l(2c+1, 2c)\leqslant 2^l$ implies

$$2c(l+2)+l+2\leqslant 2^l,$$

hence

$$c(l+2)+l-\left[\frac{l}{2}\right]=c(l+2)+\frac{l}{2}\leqslant 2^{l-1}$$

and

$$c(l+2)+\left[\frac{l}{2}\right]+1=c(l+2)+\frac{l}{2}+1\leqslant 2^{l-1}.$$

If l is odd then

$$c(l+2)+l-\left[\frac{l}{2}\right]=c(l+2)+\left[\frac{l}{2}\right]+1=c(l+2)+\frac{l}{2}+\frac{1}{2}\leqslant\frac{2^l}{2}=2^{l-1}.$$

which implies

$$\mathrm{ch}\left(c+1, 2c-\left[\frac{l}{2}\right]\right),\qquad \mathrm{ch}\left(c, 2c+\left[\frac{l}{2}\right]+1\right)\leqslant l-1$$

in this case as well.

Next we prove the lemma assuming $b=a-1\geqslant 1$, a odd and $a\leqslant 5$, i.e., for states (3, 2) and (5, 4). Consider the state (3, 2). We have $\mathrm{ch}(3, 2)=5$. The question about 2 elements of the truth-set yields states (2, 1) and (1, 4). Conditions 1 and 2 are clearly satisfied, for Condition 3 notice that $w_4(2, 1)=11$ and $w_4(1, 4)=9$, hence both are less than 2^4.

Next consider the state (5, 4). We have $\mathrm{ch}(5, 4)=6$. The question about 3 elements of the truth-set yields states (3, 2) and (2, 7). Again Conditions 1 and 2 are satisfied and for Condition 3 we notice that $w_5(2, 7)=19<w_5(3, 2)=20<2^5$.

This finishes the proof of the lemma for $b=a-1\geqslant 1$.

It remains to prove that the lemma is also valid for $b>a-1\geqslant 1$. Let $b=a-1+x$ for some $x>0$, $m=\mathrm{ch}(a, b)$ and $l=\mathrm{ch}(a, a-1)$. By the first

part of the proof there is a question about s elements of the truth-set and t elements of the lie-set which—when asked in the state $(a, a-1)$—yields states (a_1, b_1) and (a_2, b_2) satisfying the conditions of the lemma. Let $v_1 = w_{m-1}(a_1, b_1)$, $v_2 = w_{m-1}(a_2, b_2)$. In view of $m \geqslant 1$ and of Condition 3, we have $v_1, v_2 \leqslant 2^{m-1}$. Obviously $w_m(a, a-1) = v_1 + v_2$ and $w_m(a, b) = v_1 + v_2 + x$.

Let $y = \min(x, 2^{m-1} - v_1)$. The inequalities $y \geqslant 0$ and $x - y \geqslant 0$ hold. Consider the question about s elements of the truth-set and $t + y$ elements of the lie-set. Asked in the state (a, b) this question yields states $(a_1, b_1 + y)$ and $(a_2, b_2 + x - y)$. They clearly satisfy Conditions 1 and 2 of the lemma. Moreover we have:

$$w_{m-1}(a_1, b_1 + y) = v_1 + y \leqslant v_1 + 2^{m-1} - v_1 = 2^{m-1}$$

and

$$w_{m-1}(a_2, b_2 + x - y) = v_2 + x - y,$$

the latter being either v_2 (if $y = x$) or $v_2 + x - 2^{m-1} + v_1$ (if $y = 2^{m-1} - v_1$). In the first case

$$w_{m-1}(a_2, b_2 + x - y) = v_2 \leqslant 2^{m-1},$$

in the second one

$$\begin{aligned} w_{m-1}(a_2, b_2 + x - y) &= v_1 + v_2 + x - 2^{m-1} \\ &= w_m(a, b) - 2^{m-1} \leqslant 2^m - 2^{m-1} = 2^{m-1}. \end{aligned}$$

Hence those states satisfy Condition 3 as well which finishes the proof of the lemma.

We now formulate and prove the necessary and sufficient condition for the Questioner's win in the $[n, k]$ game.

THEOREM. (a) *For even n the Questioner wins the $[n, k]$ game if and only if $n(k+1) \leqslant 2^k$.*

(b) *For odd n the Questioner wins the $[n, k]$ game if and only if $n(k+1) + (k-1) \leqslant 2^k$.*

Proof. The "only if" part of (a) and (b) follows from Lemma 1. We prove the "if" part.

(a) Let $n = 2a$ and $n(k+1) \leqslant 2^k$. The first question about a elements of the truth-set in the initial state $(2a, 0)$ yields states (a, a) and (a, a) with $\mathrm{ch}(a, a) \leqslant k-1$. If $a = 1$ the Questioner wins in at most $k-1$ further questions, in view of Lemma 2 and we are done. If $a > 1$, the state (a, a) satisfies the assumptions of Lemma 3. We apply Lemma 3 repeatedly until states of the form $(1, b)$ are reached. Note that after each application the two obtained states again satisfy the assumptions in view of Condition 2. Condition 1 guarantees that states $(1, b)$ are always reached after

$t \leqslant [\lg a] + 1$ questions. In view of Condition 3 we get $\mathrm{ch}(1, b) \leqslant k - 1 - t$ for each of those states. Then by Lemma 2 the Questioner wins in at most $k - 1 - t$ questions starting from each of those states. Since he used $t + 1$ questions before, he wins the $[n, k]$ game.

(b) Let $n = 2a + 1$ and $n(k+1) + (k-1) \leqslant 2^k$. The first question about $a + 1$ elements of the truth-set in the initial state $(2a + 1, 0)$ yields states $(a + 1, a)$ and $(a, a + 1)$. Since $w_{k-1}(a, a+1) = ak + a + 1 \leqslant w_{k-1}(a+1, a) = (a+1)k + a = ((n+1)/2)k + (n-1)/2 = (n(k+1) + (k-1))/2 \leqslant 2^{k-1}$, we get

$$\mathrm{ch}(a, a+1), \mathrm{ch}(a+1, a) \leqslant k - 1.$$

The rest of the proof is exactly as in Part (a).

COROLLARY. *In the case* $n = 10^6$, *the minimal* k *for which*

$$10^6(k+1) \leqslant 2^k$$

is 25. *Hence this is the answer to Ulam's original problem: the least number* k *of yes–no questions sufficient to find an integer between* 1 *and one million, if one lie is allowed, is* $k = 25$.

Remarks. 1. The minimal number k of questions yielding the Questioner's win for odd n cannot be always obtained using the weaker condition $n(k+1) \leqslant 2^k$ good for even numbers. (i.e., the condition from point (b) cannot be replaced by the weaker one from point (a)). The example is $n = 9$ and $k = 6$ which gives

$$n(k+1) \leqslant 2^k < n(k+1) + (k-1).$$

In this case the minimal k obtained from the inappropriate formula $n(k+1) \leqslant 2^k$ applied for $n = 9$ is $k = 6$ which does not yield the Questioner's win (in view of the inequality $2^k < n(k+1) + (k-1)$).

2. For even n the least number k for which the Questioner wins the $[n, k]$ game is always the smallest of the two numbers allowed by the bounds obtained in [3] and [4]. For odd n no such property holds. If $n = 7$ both Rivest's and Spencer's results allow the minimal k to be 6 or 7 while it is actually 6. However if $n = 9$, those results allow the minimal k to be 6 or 7 as well, while it actually equals 7.

3. The Questioner's Winning Algorithm

In this section we give an algorithm yielding the Questioner's win in the $[n, k]$ game, for any natural n and the smallest k for which his winning

strategy exists. Since our algorithm follows exactly consecutive steps of the proof of the main theorem, the same proof justifies its correctness.

We first define a useful function and give a subalgorithm. Let $S = \{s_1, \ldots, s_l\}$ be a finite set of natural numbers, such that $s_1 < s_2 < \cdots < s_l$ and let $a \leqslant l$ be a natural number. Then

$$\text{chunk}(S, a) = \{s_1, \ldots, s_a\}.$$

(We let $\text{chunk}(S, 0) = \varnothing$).

The following subalgorithm RESULT describes what happens in a given state with truth-set truth and lie-set lie after the question about the first u elements of the truth-set and the first v elements of the lie-set (we note this query as ASK ($x \varepsilon$ chunk (truth, u) $\cup$ chunk (lie, v)?)). RESULT takes the values of truth, lie, u and v and returns new values of truth and lie as well as their cardinalities a and b and the character char of the new state.

Subalgorithm RESULT (truth, lie, u, v, a, b, char).

```
begin
  if answer = yes then
    begin
      lie := chunk (lie, v) ∪ truth\chunk (truth, u);
      truth := chunk (truth, u)
    end
  else
    begin
      lie := lie\chunk (lie, v) ∪ chunk (truth, u);
      truth := truth\chunk (truth, u)
    end;
  a := card (truth); b := card (lie);
  char := min{j: a*(j+1)+b ⩽ 2^j}
end;
```

Now we present the main algorithm. The variable truth denotes the truth-set and the variable lie the lie-set at a given stage of the game. a and b denote the cardinalities of those sets and char—the character of the state (a, b). We also introduce an auxiliary variable ch which denotes the character of the state $(a, a-1)$.

The first question yields states (a, a), (a, a) or $(a+1, a)$, $(a, a+1)$, depending on the parity of the integer n denoting the cardinality of the search space. Then the "while $a > 1$" loop is executed. Variables t and l denote the cardinalities of those subsets of truth and lie respectively, about

which the next question should be asked. First they are set as if the state was $(a, a-1)$ and then l is adjusted by the integer adjust in case $b > a-1$. (Here we follow the steps in the proof of Lemma 3). Note that for odd a the integer $(a-1)/2 - [\mathrm{ch}/2]$ is nonnegative for $a > 5$ and hence l is set to it before adjusting; for $a = 3$ or $a = 5$ this integer is negative which sets l to 0 according to the anlysis performed in the proof of Lemma 3.

After leaving the first loop, the state becomes $(1, b)$. The "while ($a = 1$ and $b \geqslant$ char)" loop and the "if ($a = 1$ and $b > 0$)" and "if $a = 1$" clauses coming afterwards take care of this case, following the proof of Lemma 2. Finally the last loop covers the states of the form $(0, b)$.

Algorithm STRATEGY

Input: n – the cardinality of the search space.

Output: x – the hidden number.

```
begin
  truth := {1,..., n}; lie := ∅;
  ASK (xε chunk (truth, [n/2])?);
  RESULT (truth, lie, [n/2], 0, a, b, char);
  ch := min{j: a*(j+1) + (a-1) ≤ 2^j};
  while a > 1 do
    begin
      if a is even then
        begin
          t := a/2; l := a/2
        end
      else
        begin
          t := (a+1)/2; l := max((a-1)/2 - [ch/2], 0)
        end;
      weight := t*char + l + a - t;
      adjust := min(b - (a-1), 2^(char-1) - weight);
      l := l + adjust;
      ASK (xε chunk (truth, t) ∪ chunk (lie, l)?);
      RESULT (truth, lie, t, l, a, b, char);
      ch := min{j : a*(j+1) + (a-1) ≤ 2^j}
    end;
```

```
while (a = 1 and b ≥ char) do
  begin
    ASK (x ε truth ∪ chunk (lie, ⌈(b − char + 1)/2⌉)?);
    RESULT (truth, lie, a, ⌈(b − char + 1)/2⌉, a, b, char)
  end;
if (a = 1 and b > 0) then
  begin
    ASK (x ε truth?);
    RESULT (truth, lie, a, 0, a, b, char)
  end;
if a = 1 then
  x := the unique element of truth;
while b > 1 do
  begin
    ASK (x ε chunk (lie, ⌈b/2⌉)?);
    RESULT (truth, lie, 0, ⌈b/2⌉, a, b, char)
  end;
x := the unique element of lie
end.
```

References

1. E. R. Berlekamp, Block coding for the binary symmetric channel with noiseless, delayless feedback *in* "Error-Correcting Codes," pp. 61–85, Wiley, New York, 1968.
2. B. Ravikumar and K. B. Lakshmanan, Coping with known patterns of lies in a search game, *Theoret. Comp. Sci.* **33** (1984), 85–94.
3. R. L. Rivest, A. R. Meyer, D. J. Kleitman, K. Winklmann and J. Spencer, Coping with errors in binary search procedures, *J. Comput. System Sci.* **20** (1980), 396–404.
4. J. Spencer, Guess a Number—with Lying, *Math. Mag.* **57**, No. 2 (1984), 105–108.
5. S. M. Ulam, "Adventures of a Mathematician," Scribner's, New York 1976.