Unreliable Tests

不可信赖的测试

Proposed by Stanislaw M. Ulam (1909~1984) in his autobiography "Adventures of a Mathematician" (1976).

Ulam's Game

在10^6个数中猜一数，负责回答的人可以说最多两次谎，要问多少次才可以猜出该数。（问题的形式为 $x>?$）

General Model (Group Testing).

There are n items, d defectives and at most r errors are allowed. (

Terms For any algorithm α identifying all defectives with such unreliable tests, let $N_\alpha^r(\sigma|d,n)$ denote the number of unreliable tests performed by α on the sample σ and let

$$M_\alpha^r(d,n)=\max_{\sigma\in S(d,n)} N_\alpha^r(\sigma|d,n), \text{ and}$$

$$M^r(d,n)=\min_{\alpha} M_\alpha^r(d,n).$$

Ulam's problem is to find $M^1(1,10^6)$ and $M^2(1,10^6)$.

Notations.

A "history" of GT is a sequence of tests together with outcomes.

For example, finding a number from $[1,64]$. (Answer: 45).

$([33,64], 1)$ $\quad$ $([49,64], 0)$ $\quad$ $[41,48], 1$ $\quad$ $[45,48], 1$

$(\cancel{[1,32], 0}) \longrightarrow (\cancel{[33,48], 1}) \longrightarrow (\cancel{[33,40], 0}) \longrightarrow (\cancel{[41,44], 0})$

$T([1,32]) = 0$ $\quad$ $T([33,48]) = 1$ $\quad$ $T([33,40]) = 0$ $\quad$ $T([41,44]) = 0$

$\longrightarrow ([47,48], 0) \longrightarrow (\cancel{\{45\}, 1}) \longrightarrow$ (不对：45? 0)

$T([45,46]) = 1$ $\quad$ $\boxed{T(\{45\}) = 1}$

② The above example is a reliable one.

On unreliable tests, we need to associate with a history, a state which is an ordered pair (a,b) of natural numbers. (Consider one "Lie".)

↓

a : the number of items satisfying all positive outcomes.

b : the number of items satisfying all positive outcomes except one.

For example :

$$\left.\begin{array}{l} T(\{1,2\}) = T(1,2) = 0 \\ T(2,3,4) = 1 \\ T(2,4,5) = 1 \end{array}\right)$$ In a GT of $\{1,2,3,4,5\}$ with one "Lie".

The state (a,b) associated with this history is $(a,b) = (1,3)$.

$$\begin{cases} a = |\{3,4,5\} \cap \{2,3,4\} \cap \{2,4,5\}| = |\{4\}| = 1 \\ b = |(\{3,4,5\} \cap \{2,3,4\} \cap \overset{\text{Lie}}{\boxed{\{1,3\}}}) \cup (\{3,4,5\} \cup \overset{\text{Lie}}{\boxed{\{1,5\}}} \cup \{2,4,5\}) \\ \quad \cup (\overset{\text{Lie}}{\boxed{\{1,2\}}} \cap \{2,3,4\} \cap \{2,4,5\})| = |\{3,5,2\}| = 3 \end{cases}$$

根據上述的觀察，我們可以定義以下兩個集合：

A (True-set) $= \{ v \in N \mid v$ is in all positive pool and v can be positive$\}$.

B (Lie-set) $= \{ v \in N \mid v$ is in all positive pool except one and v can be positive$\}$.

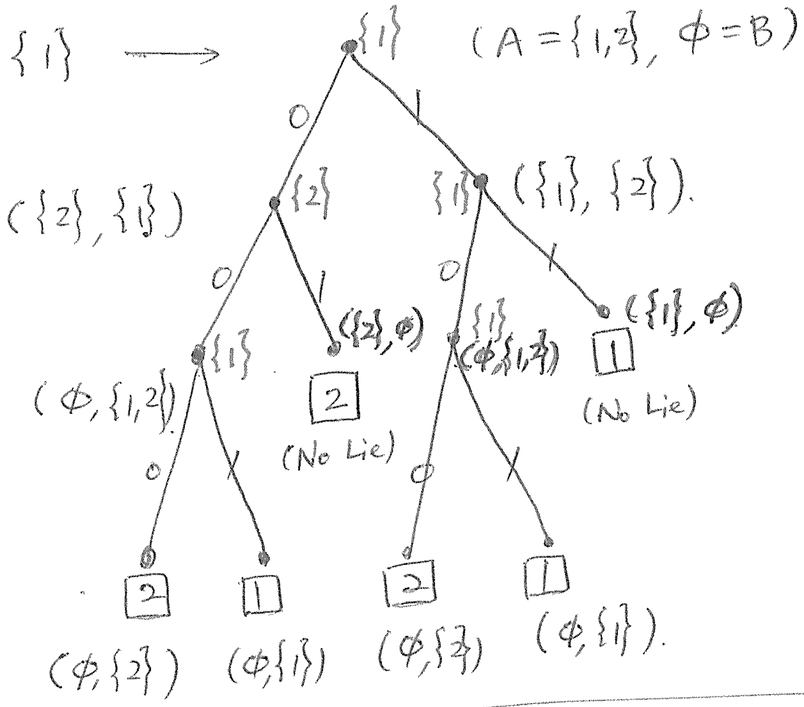
以下是一個有兩個 items 的例子，在 $\{1,2\}$ 中，1 是 defective item

(*) 如何確定 1 的確是要找的 item 呢？

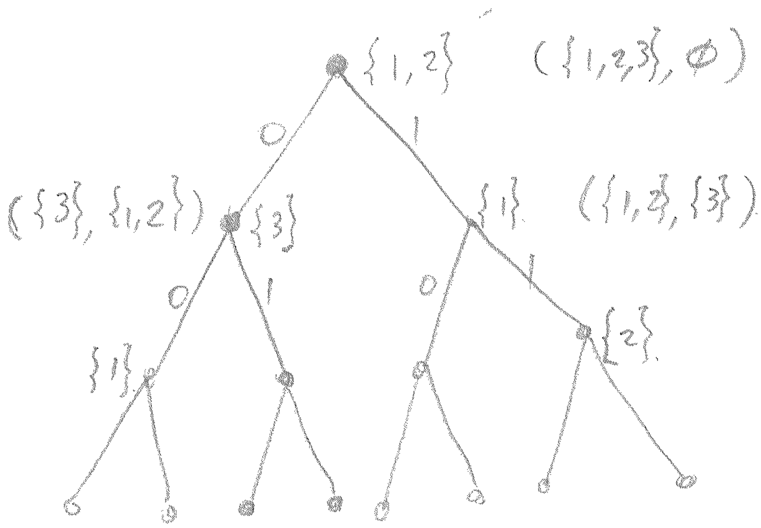
在 $A=\phi$ 時，表示 "Lie" 已經用過了，所以如果 items 的總數在 $2^{k-1} < |S| \le 2^k$ 時，至多再 k 次即可。

$S=\{1,2\}$ (已经知道有1个 defective item)

Test $\{1\}$ ⟶



$S=\{1,2,3\}$ (One defective item)



✓ A. Pelc, Solution of Ulam's problem on searching with a lie, JCTA 44 (1987), 129–140.

✓ A. Pelc, Detecting errors in searching games, JCTA 51 (1989), 43–54.

✓ A. Pelc, Prefix search with a lie, JCTA 48 (1988), 165–173.

52 JCTA
三位作者
Two lies

A. Negro and Sereno,

✓ ① Solution of Ulam's problem on binary search with three lies, JCTA 59 (1992) 149–154.

② An Ulam's searching game with three lies, Advances in Applied Mathematics, Vol. 13, Issue 4, Dec. (1992) 404–428.

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

$w_j(a,b) = a(j+1)+b$ (j questions remain to be asked.)

b：Lie set 的大小，也就是说可能产生 Lie 的次数

a：目前是 Truth set，在未来 j 次 Tests 中可能 Lie 或不 Lie 的机会，每个 item 皆为 $j+1$ 次。

$w_j(a,b) = a(j+1)+b$ 为总共可能会有的可能性.

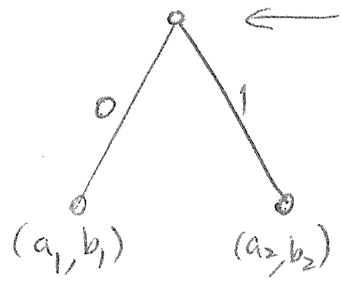
↑

Weight function

随着多次的 Test，weight 的逐渐变小；因为 Truth set 会变小，而且剩下 tests 的個数也会变小；b 可能变大。

(*) 如果有 n item，预备作 k 次 Test，则在 root 上的 weight 为 $n(k+1)$；当然如果在 k 次可以完成工作的话 $n(k+1) \leq 2^k$，否则不合乎 Information lower bound.

②



State (a, b) and test S which contains x items from A (truth set) and y items from B (Lie set).

Now, $(a_1, b_1) = (a-x, b-y+x)$ and

$(a_2, b_2) = (x, a-x+y)$

x items 不再是"positive" $\Rightarrow a_1 = a - x$.

可能 Lie 的元素增加了 x 個，而那 y 個不再是 Lie set 中的元素（因為 negative 兩次了）。

所以 $b_1 = b + x - y$.

這 x 個是原本 Truth set 的一部份，取交集就是這 x 個 items.

Test 的 outcome 是"1"說明了 y items 可以保留在 Lie set 中之外仍然有 A 中扣除 x items 的其它 items，所以 $b_2 = y + a - x$.

（B 中 y 個 items 以外的 items 在此次 Test 後基本上不會在 Lie set 中。）

所以,

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

$$w_{j-1}(a_1,b_1) = a_1(j-1+1)+b_1 = (a-x)j+b-y+x$$

$$+ \quad w_{j-1}(a_2,b_2) = a_2(j-1+1)+b_2 = xj+a-x+y$$

$$= aj+a+b$$

$$= a(j+1)+b$$

$$\Rightarrow w_j(a,b) = w_{j-1}(a_1,b_1)+w_{j-1}(a_2,b_2).$$

$\circ$ (a,b)

j tests

$w_j(a,b) = \sum w(\circ \quad \circ \quad \circ \quad \circ)$ weights here

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

If α is an algorithm which can identify the defective within k tests then in the computation tree the leaves are of states either $(1,0)$ or $(0,1)$. Therefore, their weights are either $j+1$ or 1. Now, let the length of path from the root to the leaf is $k-j$ $({}^{k}j \geq 0)$. Then, the weight of the leaf is more than $\frac{n(k+1)}{2^{k-j}}$. (尚有 j 个 tests 要做。)

所以当 $n(k+1) > 2^k$ 时，leaf 的 weight 会 $> 2^j$ 亦即 $j+1 > 2^j$ or $1 > 2^j$，均不可能。

No way to identify the defective item!

Fact

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

(2) When n is odd, if $n(k+1)+(k-1) > 2^k$, then $M'(1,n) > k$.

Proof. (1) By 4

(2) The first test must yield states (a_1, b_1), (a_2, b_2) such that

$$\max\left(w_{k-1}(a_1,b_1), w_{k-1}(a_2,b_2)\right) \geq \frac{\overset{\frac{n}{2}\cdot k + \frac{n}{2}}{n(k+1)}}{2} = \left(\frac{n+1}{2}\right)k + \frac{n-1}{2}$$

$> 2^{k-1}$.

Therefore k tests are not enough.

∎

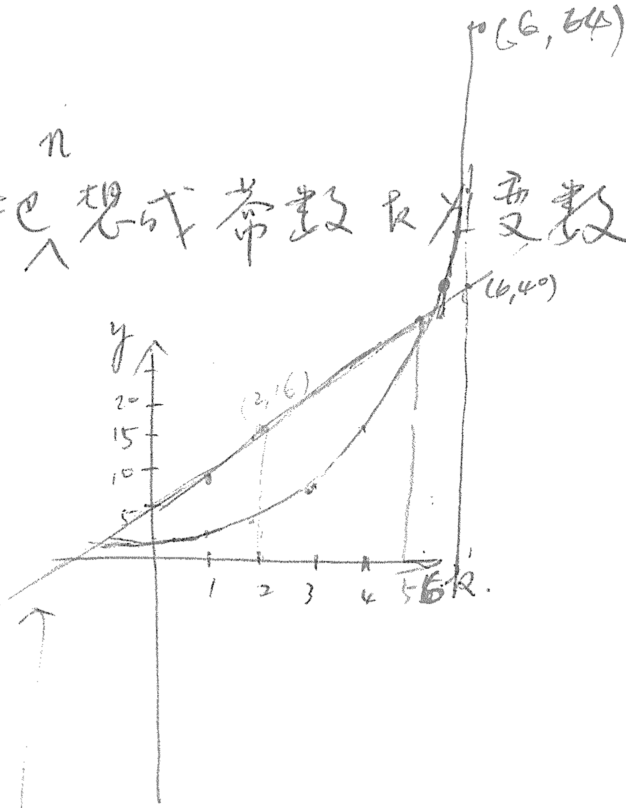
Fact

$$\begin{cases} M'(1,n) \geq \min\left\{k \mid n(k+1) \leq 2^k\right\}, & n \text{ even} \\ M'(1,n) \geq \min\left\{k \mid n(k+1)+k-1 \leq 2^k\right\}, & n \text{ is odd} \end{cases}$$

6

如果把 n 想成常數 k 為變數； $n=5$

則



$$y = 5k + 5 + (k-1)$$

$$y = 2^k$$

$\Rightarrow$ $k = 5$, $32 < 34$

$\Rightarrow$ $k = 6$ Answer!

<u>Definition</u> (Character of a state)

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

$ch(a,b) = \min\{h \mid w_h(a,b) \leq 2^h\}$. $\quad ch(1,5)=4$

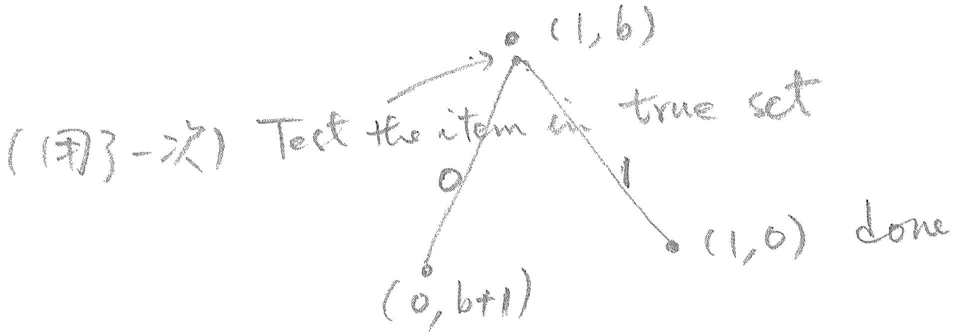
$\min\{h \mid h+1+5 \leq 2^h\} \Rightarrow h=4$

<u>Lemma</u> 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.

<u>Proof</u>. By induction on b.

Clearly, it's true if $b=1$. $\quad ch(1,1) = \min\{h \mid h+1+1 \leq 2^h\} = 2$

<u>Case 1</u> $\quad b < k$



check weight $\quad w_{k-1}(0, b+1) = b+1 \leq k \leq 2^{k-1}$

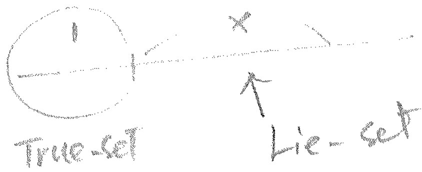
(在 True-set 為 φ 的情況下，以下的 Tests 都不可以再 Lie；所以再 $k-1$ 次就夠了！)

Case 2

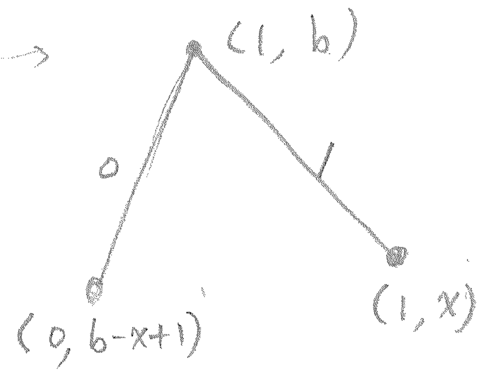
$b \geq k$

Let $x = \left\lfloor \frac{b-k+1}{2} \right\rfloor$.

Test



(唯一的 item 在 True-set 及 x 个任意 items 在 Lie-set)



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

$$= |(k+x) - (b-x+1)|$$

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

$$w_{k-1}(1,x) + w_{k-1}(0, b+1-x)$$
$$= k+x+b-x+1$$
$$= k+1+b$$
$$= w_k(1,b) \leq 2^k$$

$$\Rightarrow w_{k-1}(1,x) \leq 2^{k-1},\ w_{k-1}(0, b+1-x) \leq 2^{k-1}$$

$$ch(1,x) = \min\{h \mid w_h(1,x) \leq 2^h\} \leq k-1$$

$$ch(0, b+1-x) = \min\{h \mid w_h(0, b+1-x) \leq 2^h\} \leq k-1$$

$h+1+x \leq 2^h \Rightarrow$ $(1,x)$, $x<b$ <u>By induction</u>

$\boxed{b+1-x \leq 2^{k-1}} \Rightarrow$ $k-1$ more tests (No Lie anymore!)

So, k tests are enough!