1-1: The Counterfeit Coin Problem

Hengfeng Wei

hfwei@nju.edu.cn

Sep 10, 2019

You have eight similar coins and a beam balance.
At most one coin is counterfeit and hence underweight.
How can you detect whether there is an underweight coin,
and if so, which one, using the balance only twice?

— *E.D. Schell, 1945 (American Mathematical Monthly)*

— *Problem 1.8 of UD*

The minimum number of weighings ...

In the worst-case scenario

Decision tree

"min-max"

Put equal numbers of coins on opposite sides of the balance.
Same?

What is the First Step?

$$L = x \qquad R = x$$

Possible outcomes:

Balanced

L Rises

R Rises

Balanced: The "Standard Coin" Variation

Key point: G

L Rises: The "Labelled Coin" Variation

Key point: PH & PL & G

A Special Case of the "Labelled Coin" Variation

The counterfeit coin is known to be light.

Recursive algorithm:

$$\frac{1}{3}$$

Lower bound:

a single weighing of any sort cannot do better than trisection

The "Labelled Coin" Variation

Recursive algorithm:

> *Whenever we place coins on the scale, we must be sure to put equal number of PL (therefore PH) coins on the two sides.*

Lower bound:

> *cannot do better than in the "Light Coin" variation*

The "Labelled Coin" Variation in the 12 Coins Example

The "Standard Coin" Variation

$$M(n) = (3^n - 1)/2$$

The Weighing Algorithm (0)

Compute n, the minimum number of weighings:

$$(3^{n-1} - 3)/2 < |S| \leq (3^n - 3)/2$$

The Weighing Algorithm (1)

$$S = S_1 \cup S_2 \cup S_3$$

$$|S_1| = |S_2| \qquad |S_1 \cup S_2| \le 3^{n-1} - 1$$

$$|S_3| \le (3^{n-1} - 1)/2$$

The Weighing Algorithm (2: Balanced)

$$S_3 \cup \{\text{a standard coin}\}$$

$$S_3 = S'_1 \cup S'_2 \cup S'_3$$

$$|S_3| \leq (3^{n-1} - 1)/2$$

$$|S'_1| = |S'_2| + 1 \text{ (the standard coin)} \qquad |S_1 \cup S_2| \ldots$$

$$|S'_3| \leq (3^{n-2} - 1)/2$$

The Weighing Algorithm (2: Not Balanced)

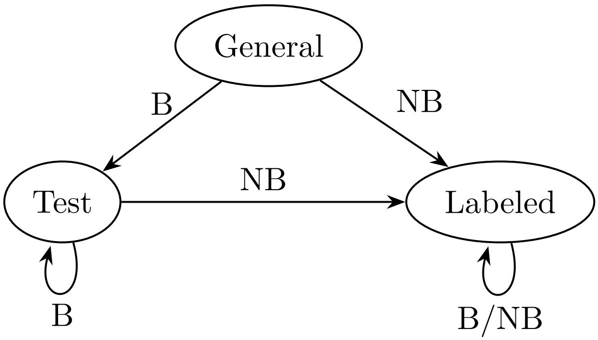
$$|S_1 \cup S_2| \le 3^{n-1} - 1$$

$$|S_1 \cup S_2| = S_1' \cup S_2' \cup S_3'$$

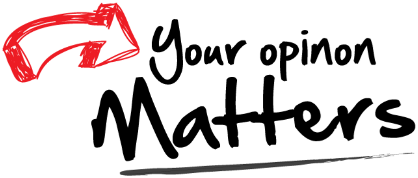
$$|S_1'| = |S_2'| \le 3^{n-2} \qquad |S_1'|_{PH}| = |S_2'|_{PL}|$$

$$|S_3'| \le 3^{n-2}$$

The Weighing Algorithm (3: Recurse)




THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn