Lecture 1 Oct. 5th.

問題

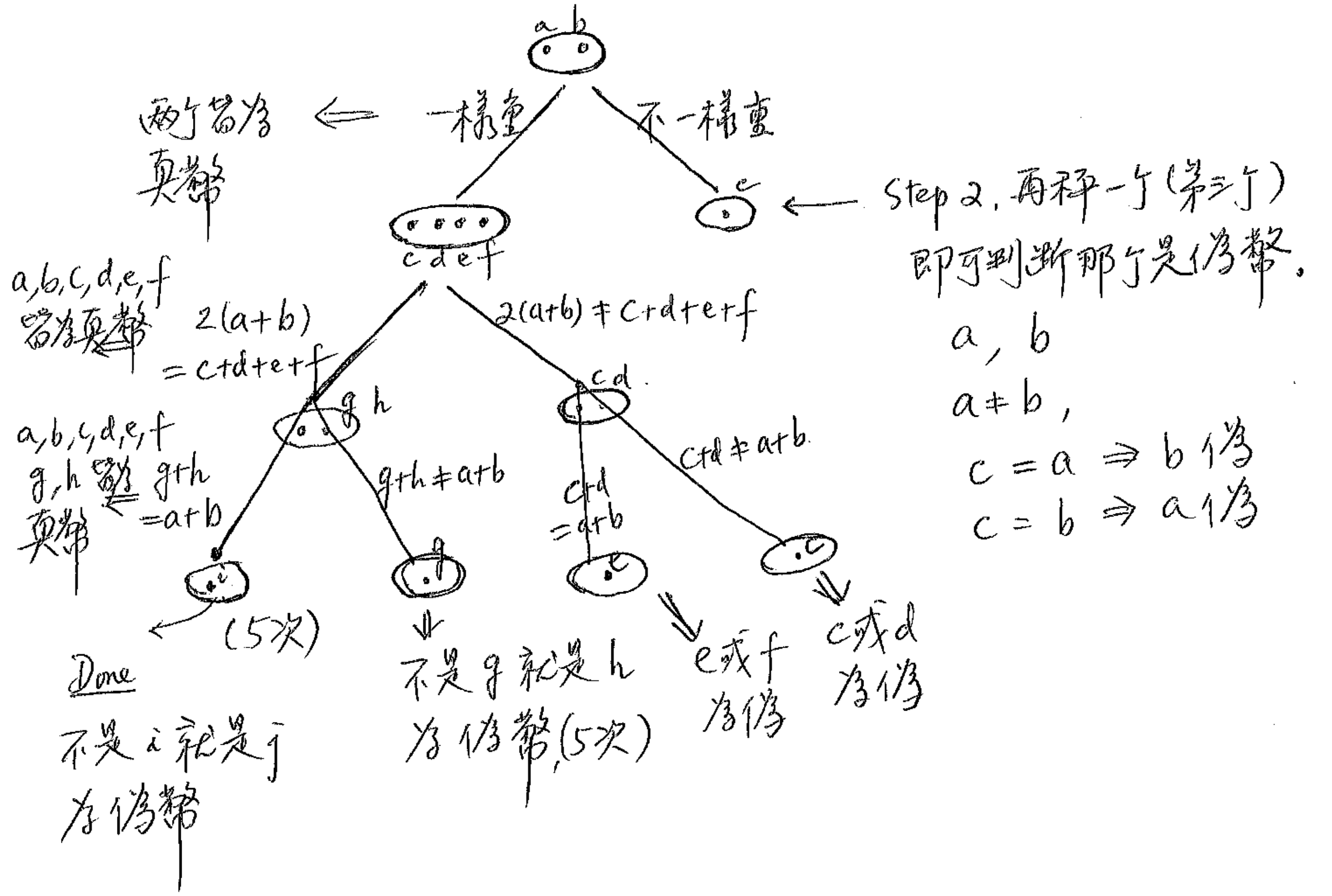
已知在10個銅幣中有一個是偽幣，它的重量與真幣不同。現在我們有一个可以量重量的儀器可以使用，那麼如何秤銅幣可以比較快找到偽幣？（偽幣可能較重也可能較輕）.

(註) 請提供一个"演算法"能正確地找出偽幣。

我們要比較那種方法在最差狀況下能以最少的秤重次數獲得答案。

(○) 兩兩比較的演算法.（5次最多）

Step 1. 任取兩个，各秤一次。（2次）



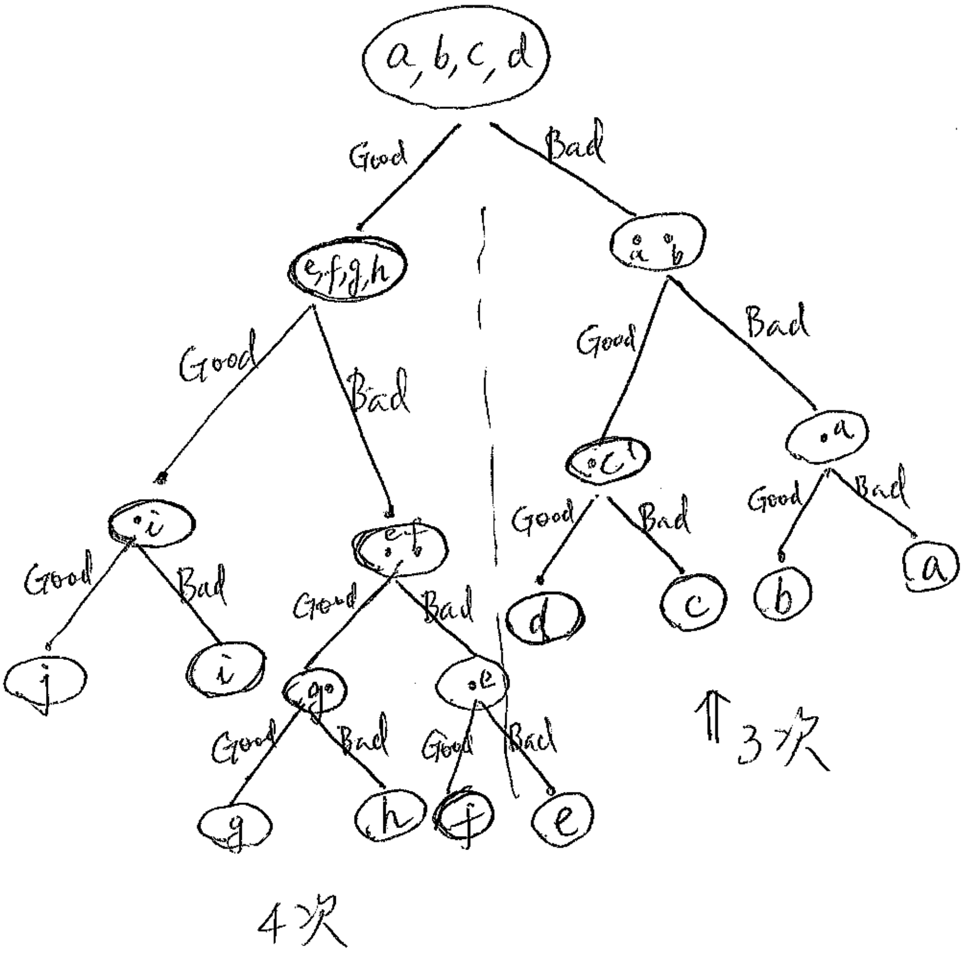
除了上述的演算法之外，我們也可以先拿出4個秤一次及另4個秤一次；接下來再從兩次秤出的結果決定後續的動作。顯然當上述兩次所得結果相同時，再一次即可決定那一個是偽幣；另一方面當兩者重量不同時就得再用三次才能決定那一個是偽幣。

問題：可以在4次完成確定偽幣的工作嗎？

如果我們更換儀器，這儀器比較精密，當你把一些硬幣投入時，它會顯示"沒有偽幣"或是"有偽幣"。

(*) 利用二分法即可在最多4次決定偽幣。

(**) 也可能全部都是真幣。



場景一

自來水廠每天輸出適當的水量到台北市的各家用戶。依照總輸出量與總收入的計算發現入不敷出的現象，也就是說水在流入用戶之前有些地方漏水；查出在什麼地方漏水非常重要，也有一定的難度。

場景二

冠狀病毒的影響之下不斷有人確診，那麼究竟是從那裏開始傳播開來的呢？如何在列人群來篩檢可以找出感染源是一個具有挑戰性的課題。

場景三

有些食物不可以同時食用，不當的組合具有對人體的很大的影响力，那麼是那些組合構成的呢？除了研究食物內在的结構之外，又如何作適當的實驗以協助判斷呢？

場景四

藥物的組合才容易產生療效，那麼如何找到這些組合呢？

場景五

人体基因序列的定序，如何進行呢？

Combinatorial Group Testing (CGT)

- Group testing is a kind of search which refers to locating certain elements (positives or defectives) of a set having a prescribed property.
- Consider a set N of n items (elements) known to contain d defectives (with prescribed properties fake coins for example). (CGT)

(*) In CGT, we make the following assumptions:

(1) The defectives look exactly like the good items.

(2) A test can be applied to an arbitrary subset S of N.

(3) There are "essentially" two outcomes: negative and positive.

(4) A negative outcome indicates that all items in S are good. This group is said to be pure. On the other hand, a positive outcome indicates that at least one the items in S is defective but not knowing which ones or how many of them are defectives. This group is said to be contaminated.

(**) If the assumptions are revised, then we have different models of CGT.

So, the plan is to find the set of "d" defectives (positives) by using as less tests as possible (in the worst case).

(•) For convenience, we call this type of group testing as (d, n)-problems, denoted by $G(d, n)$. In what follows, we shall use "positives" to replace "defectives" in terminology.

(*) By the nature of search, we are looking for good algorithms in order that we can locates all the positives by using less tests. The complexity of the algorithm will be measured in the worst case situation.

Consider an algorithm A.

- We use $M_A(d,n)$ to denote the "worst-case" number of tests required by A, namely, the maximum number of tests over all sample n points with d positives (exactly). If d is an upper bound, then we use $M_A(\bar{d},n)$.

- An algorithm is reasonable if it contains no test whose outcome can be predicted from the outcomes of other tests either conducted "previously or simultaneously".

For example, if S is tested with negative outcome, then no proper subset of S should be tested again. (Except the cases when errors are possible.)

- Define $M(d,n) = \min\limits_{A} M_A(d,n)$.

- We can also consider the sample space of a group testing problem $\mathbf{S}$.

$M_A(\mathbf{S})$, $M(\mathbf{S}) = \min\limits_{A} M_A(\mathbf{S})$.

- An algorithm A is called a minimax algorithm for $\mathbf{S}$ if $M_A(\mathbf{S}) = M(\mathbf{S})$.

Fact 1 $M(\mathbf{S}) \geq \lceil \log|\mathbf{S}| \rceil$, log stands for $\log_2$.

Proof. Since we have $|\mathbf{S}|$ sample points to start with and each test divides the sample points into two disjoint sets, the proof follows. ∎

(*) Fact 1 is the well-known "information theory bound" or simply informatic bound.

Fact 2. $S \subseteq S' \Rightarrow M(S) \le M(S')$. → Every algorithm for S' is also an algorithm for S except we may remove some "unreasonable" algorithm for S.

Fact 3. $n \le n' \Rightarrow M(d,n) \le M(d,n')$.

Fact 4. $M(1,n) = \lceil \log_2 n \rceil$.

Open problem Determine $M(2,n)$.

⊙ The fundamental group testing problem is to determine $M(d,n)$.

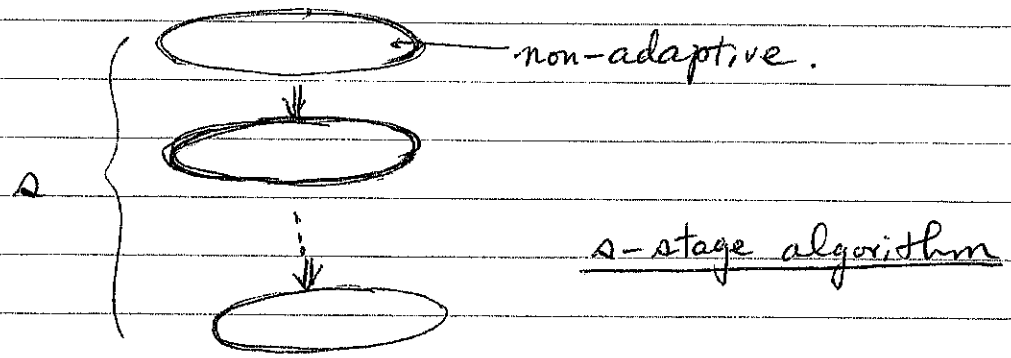
Exercise 1. Determine $M(2,6)$ and $M(\bar{2},6)$.

Fact 5. $M(\bar{d},n) \le M(d,n)+1$. [Hwang et al., 1981]

Exercise 2. Verify Fact 5.

There are two general types of group testing algorithms. In a "sequential" algorithm, the tests are conducted "one by one", and the outcomes of previous tests are assumed known at the time of determining the current test. In a "non-adaptive" algorithm, no such information is available in determining a test. Therefore, we can conduct all the tests simultaneously. (Save time!?)

A compromise between them is the so-called "multistage" algorithm: Tests are divided into several stages where the stages are considered sequential but all tests in the same stage are treated as non-adaptive.



Thus, a test can use outcomes of tests conducted in previous stages, but not those in the same stage.

=

Fact 6. (Nothing to prove!)

If an item can go through "only" one test, then individual testing is the only feasible algorithm. On the other hand, if an item can go through at most s tests, then an s-stage algorithm can be applied. See [Li] for instance. (Industrial applications)

C. H. Li, A sequential method for screening experimental variables, J. Amer. Statist. Assoc. 57 (1962), 455–477.

Group Testing with constraints

1. Each group has a size limitation, say k.

2. The items are arranged as the vertices of a "graph", only certain subgraphs (sub-groups) can be tested, for example, a path. (Graph G.T.)

3. The outcome of a test can be generalized to k-ary outcome $(0, 1, 2, \cdots, k-1^+)$, i^+ denotes at least i positives. The original outcome can be recognized as a binary outcome: $(0, 1^+)$.

   Note that a ternary (3-ary) outcome has been used in the communication (computer) which contains multiaccess channels.

4. The outcomes may contain "errors".

5. There are thresholds for being positive or negative.

6. There are items which play the role "bad" item: inhibitors.

7. Many others.

8. See Introduction for more details.