Exact Bounds for Coin Weighing by Pairwise Comparisons

Te-Sheng Tan* Dai-Yang Wu* Wing-Kai Hon

1 Introduction

There is a classic puzzle about counterfeit coin problem: a man has 12 coins among which there has a counterfeit coin, which can only be told apart by its weight. How can one tell in not more than three weighings and determine which one is a counterfeit coin [1].

The problem was so popular that have many other variants [2]. For example, the weight of counterfeit coin is heavy or light are known; given an extra coin known to be real; or even answer the question by using a spring balance i.e. a weighing device that will return the exact weight. Halbeisen [3] generize this problem when we are allowed to use more than one balances and consider more than one counterfeit coin. Although there is a lot of literature about counterfeit coin problem, but most of these solve the asymptotic bounds [4, 5, 6, 7]. Our specialty is that we solve the exact bounds.

Here we extend this question by a new direction: now we have a real coins and b counterfeit coins $(a, b > 0)$. The real coins are all the same weight and also the counterfeit coins, but two types are different weight. Each time we compare only two coins have the same weight or not. In this case, assume a, b are known. We want to study the smallest number of comparison that we can guarantee to solve these Problems:

1. find a fake coin
2. find a real coin
3. comparison two coins on the balance and one of it is fake and the other is real

We discuss the Problem 3 first, it can transfer to a new circumstance: there is an engine broken because of burned wire, and we need to find a pair of new electric wires to fix it, but all the wire are mass up, we only know that we have a positive wires and b negative wires in the beginning, each time we can choose two of them to connect to the engine and check if they are the same Electrical polarity (not work) or not (work). The object is minimize the worst case of testing times that we can fix the engine.

*Tan and Wu are co-first authors and main contributors of this work.

To solve this problem, our inisight start from an equivalent graph problem, and give a definition of foolproof schemes in Section 2. In Section 3, we show the connection between foolproof schemes and integer partition. In Section 4, we design an algorithm to solve the exact comparison times by using integer partition technique, and also speed up by cycle property. Finally, we solving the remainning Problem 1 and Problem 2 in Section 5.

2 An Equivalent Graph Problem

We can represent a testing scheme $\mathcal{S}$ for the coin-weighing problem by a graph $G_{\mathcal{S}}$ as follows:

1. It contains $\mathtt{a} + \mathtt{b}$ vertices, where each vertex represents a distinct coin.
2. Two vertices are joined by an edge if and only if $\mathcal{S}$ compares the two corresponding coins.

Note that we cannot gain extra information by comparing the same pair of coins twice; WLOG, we assume that each pair of coins are compared at most once.

Definition 1. *For a coin-weighing problem with* $\mathtt{a}$ *real coins and* $\mathtt{b}$ *fake coins, we say a testing scheme* $\mathcal{S}$ *is* foolproof *if no matter how* $\mathtt{a}$ *vertices in* $G_{\mathcal{S}}$ *are assigned as real and the remaining vertices are assigned as fake,* $G_{\mathcal{S}}$ *contains at least an* unbalanced *edge whose endpoints are with different labels; else,* $\mathcal{S}$ *is* non-foolproof*.*

Suppose that $\mathtt{a} = 3$ and $\mathtt{b} = 5$. Figure 1 shows two different foolproof schemes, while Figure 2 shows a non-foolproof scheme.

(a) A scheme with 5 comparisons (b) A scheme with 4 comparisons

Figure 1: Two foolproof schemes for $\mathtt{a} = 3$ and $\mathtt{b} = 5$. Isolated vertices are omitted for brevity.

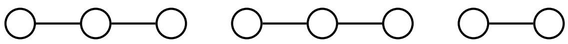

Figure 2: A non-foolproof scheme for $\mathtt{a} = 3$ and $\mathtt{b} = 5$: If the leftmost $\mathtt{a}$ vertices are assigned as real, there is no unbalanced edge.

Let $\mathcal{G}(\mathtt{a}, \mathtt{b})$ be a graph formed by the union of two cliques $K_{\mathtt{a}}$ and $K_{\mathtt{b}}$ of sizes $\mathtt{a}$ and $\mathtt{b}$, respectively. We have the following theorem.

Theorem 1. *A testing scheme* $\mathcal{S}$ *is foolproof if and only if* $G_{\mathcal{S}}$ *is not isomorphic to any subgraph of* $\mathcal{G}(\mathtt{a}, \mathtt{b})$*.*

Proof. Suppose that $G_{\mathcal{S}}$ is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a}, \mathtt{b}) = K_{\mathtt{a}} \cup K_{\mathtt{b}}$. Then, we may select an arbitrary isomorphism, and label as real for all those $\mathtt{a}$ vertices in $G_{\mathcal{S}}$ that are mapped to the vertices in $K_{\mathtt{a}}$, while label as fake for the remaining $\mathtt{b}$ vertices. Under such a labeling, there will be no unbalanced edge, so that $\mathcal{S}$ is non-foolproof.

Conversely, if $\mathcal{S}$ is non-foolproof, there exists a way to label $\mathtt{a}$ vertices in $G_{\mathcal{S}}$ as real and the remaining $\mathtt{b}$ vertices as fake so that there is no unbalanced edge. In other words, $G_{\mathcal{S}}$ can be partitioned into two sugraphs with $\mathtt{a}$ and $\mathtt{b}$ vertices, respectively; the former one is isomorphic to a subgraph of $K_{\mathtt{a}}$ and the latter one is isomorphic to a subgraph of $K_{\mathtt{b}}$, so that $G_{\mathcal{S}}$ is isomorphic to a subgraph of $\mathcal{G}(\mathtt{a}, \mathtt{b})$. $\square$

Definition 2. *For a coin-weighing problem with* $\mathtt{a}$ *real coins and* $\mathtt{b}$ *fake coins, we say a foolproof testing scheme* $\mathcal{S}$ *is* optimal *if it requires the minimal number of comparisons in the worst case; the minimal number of comparisons is denoted by* $\tau(\mathtt{a},\mathtt{b})$.

Remark. By definition, $\tau(\mathtt{a},\mathtt{b}) = \tau(\mathtt{b},\mathtt{a})$.

The target of this paper is to design optimal foolproof testing scheme so that by comparing coins according to the scheme, we are guaranteed to compare a real coin with a fake coin using the fewest comparisons in the worst case. By Theorem 1, it is equivalent to finding a graph G, with the fewest number of edges, that is not isomorphic to any subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$.

2.1 Lower Bound

Theorem 2. *Any graph with* $n \leq \mathtt{a}+\mathtt{b}$ *vertices and* $m \leq \lfloor(\mathtt{a}+\mathtt{b})/2\rfloor - 1$ *edges is always isomorphic to some subgraph of* $\mathcal{G}(\mathtt{a},\mathtt{b})$.

Proof. Without loss of generality, we assume that $\mathtt{a} \leq \mathtt{b}$. We shall prove this theorem by induction on the sum $\mathtt{a}+\mathtt{b}$.

(Basis Case:) If $\mathtt{a} = \mathtt{b} = 1$, then $\lfloor(\mathtt{a}+\mathtt{b})/2\rfloor - 1 = 0$. Any graph with $n \leq 2$ vertices and $m \leq 0$ edges (i.e., no edges) is always isomorphic to some subgraph of $\mathcal{G}(1,1)$.

(Inductive Case:) Suppose that the theorem holds for all $\mathtt{a}+\mathtt{b} \leq k$. Our target is to show that the theorem also holds for the case $\mathtt{a}+\mathtt{b} = k+1$ with $\mathtt{a} \leq \mathtt{b}$. Consider a graph G with $n \leq k+1$ vertices and with $m \leq \lfloor(k+1)/2\rfloor - 1$ edges.

1. If G is connected, then $n \leq m+1 \leq (k+1)/2 \leq b$, which is isomorphic to some subgraph of $K_{\mathtt{b}}$, and thus isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$.

2. Otherwise, G is not connected. If G has no edges, then G is obviously isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$ since G has at most $\mathtt{a}+\mathtt{b}$ vertices. Else, let C be the connected component of G with the largest number n' of vertices (so that $n' \geq 2$. Then, the number of edges in C is at least $n'-1$. To complete the proof, it is sufficient to show that $G - C$ is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b}-n')$, as we can map the vertices of C to an arbitrary set of n' vertices in the $K_{\mathtt{b}}$ component of $\mathcal{G}(\mathtt{a},\mathtt{b})$.

   The number of vertices in $G - C$ is $k+1-n' = \mathtt{a}+(\mathtt{b}-n')$, and the number of edges of $G - C$ is at most

   $$\begin{aligned} m-(n'-1) \leq \lfloor(k+1)/2\rfloor - n' &= \lfloor(k+1)/2 - n'\rfloor \\ &= \lfloor(k+1-n')/2 - n'/2\rfloor \\ &\leq \lfloor(k+1-n')/2 - 1\rfloor \\ &= \lfloor(k+1-n')/2\rfloor - 1 = \lfloor(\mathtt{a}+(\mathtt{b}+n'))/2\rfloor - 1. \end{aligned}$$

   By induction hypothesis, $G-C$ is isomorphic to a subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b}-n')$, and consequently G is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$.

In both cases, G is isomorphic to a subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$. This completes the proof of the induction case, so that by the principle of mathematical induction, the theorem follows. □

Immediately, we have the following corollaries.

Corollary 1. *For any* $\mathtt{a}$ *and* $\mathtt{b}$, $\tau(\mathtt{a},\mathtt{b}) \geq \lfloor(\mathtt{a}+\mathtt{b})/2\rfloor$.

Proof. The corollary is a direct consequence of Theorems 1 and 2. □

Corollary 2. *When* $\mathtt{a}$ *and* $\mathtt{b}$ *are both odd,* $\tau(\mathtt{a},\mathtt{b}) = (\mathtt{a}+\mathtt{b})/2$*.*

Proof. Consider a testing scheme $\mathcal{S}$ that partitions the coins into $(\mathtt{a}+\mathtt{b})/2$ pairs, and then compares each pair of coins. The corresponding graph $G_{\mathcal{S}}$ consists of $(\mathtt{a}+\mathtt{b})/2$ disjoint edges (as in Figure 1(b)), and is not isomorphic to any subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$. Thus, $G_{\mathcal{S}}$ is foolproof, and $\tau(\mathtt{a},\mathtt{b}) \leq (\mathtt{a}+\mathtt{b})/2$. On the other hand, by Corollary 1, we have $\tau(\mathtt{a},\mathtt{b}) \geq (\mathtt{a}+\mathtt{b})/2$ for odd $\mathtt{a}$ and odd $\mathtt{b}$. The corollary thus follows. □

Corollary 3. *Consider a testing scheme* $\mathcal{S}$ *whose corresponding graph* $G_{\mathcal{S}}$ *forms a path with* $\max(\mathtt{a},\mathtt{b})+1$ *nodes (as in Figure 1(a)). Then,* $\mathcal{S}$ *is foolproof, and it uses at most twice the number of comparisons of any optimal foolproof scheme.*

Proof. The graph $G_{\mathcal{S}}$ is not isomorphic to any subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$, so that it is foolproof. The number of comparisons by $\mathcal{S}$ is

$$\begin{array}{rclcl} \max(\mathtt{a},\mathtt{b}) & \leq & \mathtt{a}+\mathtt{b}-1 & = & 2\times((\mathtt{a}+\mathtt{b}-1)/2) \\ & \leq & 2\times\lfloor(\mathtt{a}+\mathtt{b})/2\rfloor & \leq & 2\times\tau(\mathtt{a},\mathtt{b}), \end{array}$$

where the last inequality comes from Corollary 1. The corollary thus follows. □

3 From Graph to Integer Partition

This section shows an interesting connection between designing an optimal foolproof scheme and searching for an integer partitioning of some special form. We begin with the following simple lemma, which is crucial in establishing the connection.

Lemma 1. *If* $\mathcal{S}$ *is an optimal foolproof scheme, then its corresponding graph* $G_{\mathcal{S}}$ *does not contain any cycle.*

Proof. Suppose on the contrary that $G_{\mathcal{S}}$ does. Then, we can obtain a graph G' by removing an edge from some cycle in $G_{\mathcal{S}}$. If G' is not be isomorphic to any subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$, by Theorem 1, this would imply G' corresponds to a foolproof scheme; furthermore, such a scheme performs fewer comparisons than $\mathcal{S}$, so that $\mathcal{S}$ is not optimal. Otherwise, if G' is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$, then $G_{\mathcal{S}}$ would also be isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$ (using the same mapping between the vertices), so that $\mathcal{S}$ is not foolproof by Theorem 1. Contradiction occurs in both cases, and the lemma thus follows. □

Based on the above lemma, the graph $G_{\mathcal{S}}$ corresponding to an optimal foolproof scheme $\mathcal{S}$ must be a *forest*. Indeed, we may observe that the *shape* of each connected tree is not important: Precisely, for each tree T in $G_{\mathcal{S}}$ that connects some set U of vertices, we can replace T by any other tree that connects the vertices in U, and the resulting scheme remains foolproof.[1] Also, after replacing T, the new scheme remains optimal as it uses the same number of comparisons. This naturally implies that an optimal foolproof scheme is related to some kind of *partitioning* of the integer $\mathtt{a}+\mathtt{b}$. In the following, we shall unveil the property of such a partitioning. We first define a related concept.

[1] If not, the latter graph is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$, but then $G_{\mathcal{S}}$ would also be isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$ under the same vertex mapping.

Definition 3. *Let P be a multiset of positive integers. We say P* avoids *a positive integer x if for any subset $P' \subseteq P$, the sum of all integers in P' is not equal to x.*

Let $P = \{p_1, p_2, \ldots, p_{|P|}\}$ be a multiset of positive integers whose sum is $\mathtt{a}+\mathtt{b}$. In other words, P forms an integer partition of $\mathtt{a}+\mathtt{b}$. We say a testing scheme $\mathcal{S}$ *corresponds to* P if $G_{\mathcal{S}}$ is a forest whose trees have sizes $p_1, p_2 \ldots, p_{|P|}$, respectively. Then, we have the following theorem.

Theorem 3. *Let $\mathcal{S}$ be a testing scheme whose corresponding graph $G_{\mathcal{S}}$ is a forest. Then $\mathcal{S}$ is foolproof if and only if $\mathcal{S}$ corresponds to a partition P of the integer $\mathtt{a}+\mathtt{b}$ that avoids $\mathtt{a}$.*

Proof. We prove the necessary and sufficient conditions separately, each by contradiction.

($\Rightarrow$) Suppose that $\mathcal{S}$ is foolproof. Assume on the contrary that P does not avoid $\mathtt{a}$, then we can partition P into two subsets $P_{\mathtt{a}}$ and $P_{\mathtt{b}}$ such that the sum of integers in $P_{\mathtt{a}}$ is equal to $\mathtt{a}$ (and thus, the sum of integers in $P_{\mathtt{b}}$ is $\mathtt{b}$). Then, consider those trees in $G_{\mathcal{S}}$ with sizes $p \in P_{\mathtt{a}}$, they together would be isomorphic to some subgraph of $K_{\mathtt{a}}$; similarly, the remaining trees in $G_{\mathcal{S}}$ would be isomorphic to some subgraph of $K_{\mathtt{b}}$. This implies that $G_{\mathcal{S}}$ is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$, so that $\mathcal{S}$ is not foolproof by Theorem 1. A contradition occurs.

($\Leftarrow$) Suppose that P avoids $\mathtt{a}$. Assume on the contrary that $\mathcal{S}$ is not foolproof. By Theorem 1, $G_{\mathcal{S}}$ is isomorphic to some subgraph of $\mathcal{G}(\mathtt{a},\mathtt{b})$. Consider a particular isomorphism f. Note that the number of vertices of both $G_{\mathcal{S}}$ and $\mathcal{S}(\mathtt{a},\mathtt{b})$ are the same, so that f is a bijection between the two vertex sets. Based on f, we can partition the vertices in $G_{\mathcal{S}}$ into two groups, where the first group contains those who are mapped to vertices in the $K_{\mathtt{a}}$ component of $\mathcal{G}(\mathtt{a},\mathtt{b})$, and the second one contains those remaining vertices. Also, all vertices from the same tree in $G_{\mathcal{S}}$ must be in the same group. By focussing on those trees whose vertices are mapped to the first group, their sizes adds up to $\mathtt{a}$, so that P does not avoid $\mathtt{a}$. A contradiction occurs. $\square$

For instance, although the scheme corresponding to Figure 2 is not foolproof when $\mathtt{a} = 3$ and $\mathtt{b} = 5$, it is foolproof when $\mathtt{a} = 1$ and $\mathtt{b} = 7$, or when $\mathtt{a} = 4$ and $\mathtt{b} = 4$. Also, we have the following corollary.

Corollary 4. *Let $\mathcal{S}$ be an optimal testing scheme. Then, $\mathcal{S}$ corresponds to a partition $P_{\max}$ of $\mathtt{a}+\mathtt{b}$, with the maximal number of parts, that avoids $\mathtt{a}$; furthermore, $\tau(\mathtt{a},\mathtt{b}) = \mathtt{a}+\mathtt{b}-|P_{\max}|$, where the notation $|P|$ denotes the number of parts in a partition P.*

Proof. By Lemma 1, $G_{\mathcal{S}}$ is a forest. Then, by Theorem 3, $\mathcal{S}$ corresponds to a partition P of $\mathtt{a}+\mathtt{b}$ that avoids $\mathtt{a}$. Furthermore, the number of comparisons performed by $\mathcal{S}$ is exactly $\mathtt{a}+\mathtt{b}-|P|$. As $\mathcal{S}$ is optimal, this implies $\mathcal{S}$ minimizes the number of comparisons, so that the corresponding partition P maximizes the number of parts. The corollary thus follows. $\square$

Corollary 5. *For any $\mathtt{b} \geq 1$,*

$$\tau(2,\mathtt{b}) = \mathtt{b}+2-\left\lfloor \frac{\mathtt{b}+2}{3} \right\rfloor.$$

Proof. For any partition P of $2+\mathtt{b}$ that avoids 2, P contains at most one 1 and no 2s, so that

$$|P| \;\leq\; \max\left\{1+\frac{\mathtt{b}+1}{3},\;\; \frac{\mathtt{b}+2}{3}\right\} \;\leq\; 1+\frac{\mathtt{b}+1}{3} \;=\; \frac{\mathtt{b}+4}{3}.$$

Furthermore, since $|P|$ is an integer, the above inequality implies that

$$|P| \;\leq\; \left\lceil \frac{\mathtt{b}+4}{3} \right\rceil \;=\; \left\lfloor \frac{\mathtt{b}+2}{3} \right\rfloor.^{2}$$

[2]The equality follows from the fact: for any integer n and any positive integer m, $\lceil n/m \rceil = \lfloor (n+m-1)/m \rfloor$.

Thus by Corollary 4, we have $\tau(2, \mathtt{b}) \geq \mathtt{b} + 2 - \lfloor(\mathtt{b}+2)/3\rfloor$.

However, by setting P to be

- $P = \{3, 3, \ldots\}$ (where ... represents trailing 3s) for $\mathtt{b} + 2 \equiv 0 \pmod 3$, or
- $P = \{4, 3, \ldots\}$ (where ... represents trailing 3s) for $\mathtt{b} + 2 \equiv 1 \pmod 3$, or
- $P = \{1, 4, 3, \ldots\}$ (where ... represents trailing 3s) for $\mathtt{b} + 2 \equiv 2 \pmod 3$,

each partition P is a partition of $2+\mathtt{b}$ that avoids 2, and with $|P| = \lfloor(\mathtt{b}+2)/3\rfloor$. By Corollary 4, this implies that $\tau(2, \mathtt{b}) \leq \mathtt{b} + 2 - \lfloor(\mathtt{b}+2)/3\rfloor$. As $\tau(2, \mathtt{b})$ is now upper-bounded and lower-bounded by the same desired quantity, the corollary thus follows. □

4 Computing $\tau(\mathtt{a}, \mathtt{b})$

To find optimal, we enumerate all partitions of $\mathtt{a}+\mathtt{b}$. Then checking P's avoidence, we solve this as below. This idea is same as solving discrete knapsack problem by DP. This checking runs in $O(\mathtt{ab})$ for a particular partition.

Algorithm 1 Find optimal foolproof scheme

```
procedure FIND_OPTIMAL_FOOLPROOF_SCHEME(a, b)            ▷ return a OFS
    R = {a + b}
    for each partition P of the number a + b which avoids a do
        if Check_avoidence(P, a) then
            continue
        if |P| > |R| then
            R = P
    return R
procedure CHECK_AVOIDENCE(P, a)                          ▷ Whether P avoids a
    S = {0}
    for each number pi ∈ P do                            ▷ O(a + b)times
        S = S ∪ S+̣pi                                     ▷ This cost O(a)
    return a ∉ S
```

Denote the number of partitions of n as $p(n)$, the number of partitions avoids a of n as $p_{aoivds\ \mathtt{a}}(n)$

Time Complexity:

$$O(\mathtt{ab} \times p(\mathtt{a}+\mathtt{b})) = O(\mathtt{ab}\frac{exp(\sqrt{\frac{2(\mathtt{a}+\mathtt{b}))}{3}})}{\mathtt{a}+\mathtt{b}})$$

We can improve this a little by enumerating only partitions which avoids $\mathtt{a}$.

Time Complexity:

$$O(a \times p_{aoivds\ \mathtt{a}}(\mathtt{a}+\mathtt{b})) \leq O(\mathtt{a} \times p(\mathtt{a}+\mathtt{b})) \leq O(\mathtt{a}\frac{exp(\sqrt{\frac{2(\mathtt{a}+\mathtt{b}))}{3}})}{\mathtt{a}+\mathtt{b}})$$

Remark. *We shall enumerate partitions by Depth-First Search, keep avoidence set(S) when enumerating.*

Algorithm 2 Find optimal foolproof scheme

procedure FIND_OPTIMAL_FOOLPROOF_SCHEME($\mathtt{a}, \mathtt{b}$) ▷ return a OFS
    $R = \{\mathtt{a} + \mathtt{b}\}$
    for each partition P of the number $\mathtt{a} + \mathtt{b}$ which avoids $\mathtt{a}$ **do**
        if $|P| > |R|$ **then**
            $R = P$
    output R

4.1 Results

$\rho(\mathtt{a}, \mathtt{b})$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10
2	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6	7	7	7	8
3	2	2	3	3	4	3	5	4	6	5	7	6	8	7	9	8	10	9	11	10
4	2	2	3	4	4	4	4	4	5	5	5	5	6	6	6	7	7	7	8	8
5	3	3	4	4	5	5	6	5	7	5	8	6	9	7	10	8	11	9	12	10
6	3	3	3	4	5	6	6	6	6	6	6	6	7	7	7	7	7	7	7	8
7	4	3	5	4	6	6	7	7	8	7	9	7	10	7	11	8	12	9	13	10
8	4	4	4	4	5	6	7	8	8	8	8	8	8	8	8	8	9	9	9	10
9	5	4	6	5	7	6	8	8	9	9	10	9	11	9	12	9	13	9	14	10
10	5	4	5	5	5	6	7	8	9	10	10	10	10	10	10	10	10	10	10	10
11	6	5	7	5	8	6	9	8	10	10	11	11	12	11	13	11	14	11	15	11
12	6	5	6	5	6	6	7	8	9	10	11	12	12	12	12	12	12	12	12	12
13	7	5	8	6	9	7	10	8	11	10	12	12	13	13	14	13	15	13	16	13
14	7	6	7	6	7	7	7	8	9	10	11	12	13	14	14	14	14	14	14	14
15	8	6	9	6	10	7	11	8	12	10	13	12	14	14	15	15	16	15	17	15
16	8	6	8	7	8	7	8	8	9	10	11	12	13	14	15	16	16	16	16	16
17	9	7	10	7	11	7	12	9	13	10	14	12	15	14	16	16	17	17	18	17
18	9	7	9	7	9	7	9	9	9	10	11	12	13	14	15	16	17	18	18	18
19	10	7	11	8	12	7	13	9	14	10	15	12	16	14	17	16	18	18	19	19
20	10	8	10	8	10	8	10	10	10	10	11	12	13	14	15	16	17	18	19	20

$\tau(\mathtt{a},\mathtt{b})$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11
2	2	2	3	4	4	5	6	6	7	8	8	9	10	10	11	12	12	13	14	14
3	2	3	3	4	4	6	5	7	6	8	7	9	8	10	9	11	10	12	11	13
4	3	4	4	4	5	6	7	8	8	9	10	11	11	12	13	13	14	15	15	16
5	3	4	4	5	5	6	6	8	7	10	8	11	9	12	10	13	11	14	12	15
6	4	5	6	6	6	6	7	8	9	10	11	12	12	13	14	15	16	17	18	18
7	4	6	5	7	6	7	7	8	8	10	9	12	10	14	11	15	12	16	13	17
8	5	6	7	8	8	8	8	8	9	10	11	12	13	14	15	16	16	17	18	18
9	5	7	6	8	7	9	8	9	9	10	10	12	11	14	12	16	13	18	14	19
10	6	8	8	9	10	10	10	10	10	10	11	12	13	14	15	16	17	18	19	20
11	6	8	7	10	8	11	9	11	10	11	11	12	12	14	13	16	14	18	15	20
12	7	9	9	11	11	12	12	12	12	12	12	12	13	14	15	16	17	18	19	20
13	7	10	8	11	9	12	10	13	11	13	12	13	13	14	14	16	15	18	16	20
14	8	10	10	12	12	13	14	14	14	14	14	14	14	14	15	16	17	18	19	20
15	8	11	9	13	10	14	11	15	12	15	13	15	14	15	15	16	16	18	17	20
16	9	12	11	13	13	15	15	16	16	16	16	16	16	16	16	16	17	18	19	20
17	9	12	10	14	11	16	12	16	13	17	14	17	15	17	16	17	17	18	18	20
18	10	13	12	15	14	17	16	17	18	18	18	18	18	18	18	18	18	18	19	20
19	10	14	11	15	12	18	13	18	14	19	15	19	16	19	17	19	18	19	19	20
20	11	14	13	16	15	18	17	18	19	20	20	20	20	20	20	20	20	20	20	20

Previously, to find OTS, we enumerate all partitions of $\mathtt{a}+\mathtt{b}$. It cost much time. Discovering from tables in Results 3.3. We found the cycle property. Take $\mathtt{a}=5$ as example.

$\tau(\mathtt{a},\mathtt{b})$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	3	4	4	5	5	6	6	8	7	10	8	11	9	12	10	13	11	14	12	15

From $\mathtt{b}\geq 9$, $\tau(\mathtt{a},\mathtt{b})$ forms cycle of length 2. In general, when $\mathtt{b}$ is large enough $\tau(\mathtt{a},\mathtt{b})$ would forms into cycles. Using this property to improve our algorithm for the case $\mathtt{b}$ is much larger than $\mathtt{a}$. The improved algorithm runs in time complexity with no $\mathtt{b}$.

Definition 4. *d is the smallest non-factor of* $\mathtt{a}$

Lemma 2. *$\rho(a,b)$ has lower-bound $\lceil \frac{b}{d} \rceil$*

Proof. To proof a lower-bound, we show a case whose number of partitions$=\lceil \frac{b}{d} \rceil$ The scheme is trivial, divide $a+b$ into $\lceil \frac{b}{d} \rceil - 1$ instances of d and a $a+b-d\lceil \frac{b}{d} \rceil - 1$ (denoted by s)

Then we shall prove the scheme avoids $\mathtt{a}$ by contradiction.There are two cases. If the subset contains s but $s>a$. If the subset doesn't contains s, it contains only several 'd's, but $d \not| a$. Both cases raise contradiction. □

We shall setup notations before showing the folloing lemmas. For a multiset S, We denote the number of i in S as X_i, that is, we have X_i 'i'(s) in S and $X=\sum_i X_i$. S_k to denote the first k elements(e.g. prefix) of S. S_0 is the empty set.

To avoid ambiguity with union operation, we use $S \dot{+}$ a number q to denote the result multiset that adding q to each element to multiset S.

We define set $U(S)$ for a multiset s as subset-sums$\leq a$ of S. For example, $\mathtt{a}=5$, $U(1,3,3)=$ ignore$_{>5}(\{0,1,3,4,6,7\})=\{0,1,3,4\}$. Note that for any S, $U(S)$ always include zero.

Lemma 3. *If a multiset S avoids a and composed only by factors of a, $|S|<a$*

Proof. We shall prove this by contradiction

We divide the proof into two parts.

(First Part)

$$\forall k, |U(S_{k+1})| > |U(S_k)| \tag{1}$$

For simplisty we Denote $U(S_{k+1})$ as u' and $U(S_k)$ as u, the k+1th element of S as w. Note that since ignore$_{>\mathtt{a}}(u \dot{+} w) \subseteq u'$

$$\forall i \geq w, i \in u \implies i+w \in u' \tag{2}$$

We shall prove the statement 1 by contradiction, if $|u'|=|u|$ for some k, then since $u \subseteq u'$, $u=u'$. And from statement 2, we get

$$\forall i \geq w, i \in u \implies i+w \in u' \implies i+w \in u$$

Since U always include 0, and $0 \in u \implies tw \in u$ for any t. But w is a factor of $\mathtt{a}$, thus $\mathtt{a} \in u$, contradicts S avoids $\mathtt{a}$

(Second Part) We discover $U(S_0)$ to $U(S_{|S|})$, since $U(S_0)=\{0\}$, each time U at least a new element. If $|S| \geq \mathtt{a}$, then $U(S)$ must be $\{0,1,...,\mathtt{a}\}$, contradicts S avoids $\mathtt{a}$

□

Theorem 4. $\forall \mathtt{b} > d(d+1)(\mathtt{a}-1)+(\mathtt{a}-(\mathtt{a} \mod d)$, *an optimal partition contains at least* $\lfloor \frac{\mathtt{a}}{d} \rfloor$ *instances of d*

Proof. We shall prove by contradiction. Let S is a partition of $\mathtt{a}+\mathtt{b}$ avoids $\mathtt{a}$.

Assume $X_d < \lfloor \frac{\mathtt{a}}{d} \rfloor$ then

$$X = \sum_{i<d} X_i + X_d + \sum_{i>d} X_i$$

Since $\sum_i iX_i = \mathtt{a}+\mathtt{b}$, to maximize X, we use as much small number as possible. Hence

$$\begin{aligned} X &\leq \mathtt{a} - 1 + \lfloor \frac{\mathtt{a}}{d} \rfloor + \lfloor \frac{\mathtt{b} - (\mathtt{a} - (\mathtt{a} \mod d))}{d+1} \rfloor \\ &\leq \mathtt{a} - 1 + \frac{\mathtt{a}}{d} + \frac{\mathtt{b} - (\mathtt{a} - (\mathtt{a} \mod d))}{d+1} \end{aligned}$$

$\forall \mathtt{b} > d(d+1)(\mathtt{a}-1) + (\mathtt{a} - (\mathtt{a} \mod d))$,

$$X < \mathtt{a} + \lfloor \frac{\mathtt{a}}{d} \rfloor + \mathtt{a}d - d \leq \lceil \frac{\mathtt{b}}{d} \rceil$$

which is not optimal from lemma **??**.A contradiction occurs. □

Theorem 5. $\forall \mathtt{b} > d(d+1)(\mathtt{a}-1) + (\mathtt{a} - (\mathtt{a} \mod d)$*,* $\rho(\mathtt{a}, \mathtt{b}+d) = \rho(\mathtt{a}, \mathtt{b}) + 1$

Proof. $b > d(d+1)(a-1) + (a - (a \mod d)$, let S be a optimal partition of a+b avoids a, $S' = S+$'d' which is a partition of a+b+d.

We divide the proof into avoidance part and optimal part

(Avoidance) prove by contradiction

If S' has subset-sum $\mathtt{a}$, S must has subset-sum $\mathtt{a}$ or $\mathtt{a} - d$. The former is not possible from definition. If the later happens, S has subset-sum $\mathtt{a}$ since it has at least $\lfloor \frac{a}{d} \rfloor$ instances of 'd', and $\mathtt{a} - d$ couldn't has all of them. This contradicts definition. Thus, S' avoids $\mathtt{a}$.

(Optimal) prove by contradiction

If S' is not optimal there exist an optimal partition P' that $|P'| > |S'|$ and has no subset-sum=a. P' has at least one d, let $P = P' - d$(note $sum(P) = sum(S)$) then P has no subset-sum=a and $|P| = |P'| - 1 > |S'| - 1 = |S|$, which contradicts S is optimal. Thus, S' is optimal. Thus the theorem holds. □

Corollary 6. *for a constant* $\mathtt{a}$*,* $\rho(\mathtt{a}, \mathtt{b})$ *forms cycle of length* d *if* $\mathtt{b}$ *is sufficiently large*

4.2 Algorithm using cycle property

Algorithm 3 Find optimal foolproof scheme using cycle property

```
K = a + b − kd > d(d + 1)a                                    ▷ choose maximal k, e.g. minimal K
procedure FIND_OPTIMAL_FOOLPROOF_SCHEME(a, b)                             ▷ return a OFS
    R =Find_optimal_foolproof_scheme^2(a, b − Kd)     ▷ This procedure is from algorithm 2
    return R+k instances of 'd'
```

check P's avoidence by Dynamic-Programming as solving discrete knapsack problem, keep avoidence vector when enumerating.

Time Complexity:

$$O(k+a\times p_{aoivds\ \mathtt{a}}(K)) = O(\mathtt{a}\times p(K)) = O(\mathtt{a}\frac{exp(\sqrt{\frac{2(1+d(d+1)(\mathtt{a}-1)+\mathtt{a})}{3}})}{1+d(d+1)(\mathtt{a}-1)+\mathtt{a}}) = O(\frac{exp(\sqrt{\frac{2(1+d(d+1)(\mathtt{a}-1)+\mathtt{a})}{3}})}{d^2})$$

5 extended problem

We shall discuss several problems simular to the foolproof problem in section 1. These are called inference problems. For convenience, we assume fake coins are minor in the whole coins.(e.g. ($\mathtt{a} \leq \mathtt{b}$) In the same way in Section 2, we present a scheme as a graph. There are four inference problems: infering a fake coin,a real coin, real-fake pair, a fake coin and a real coin. We $-$, $+$, $\pm$, $+-$ superscription to denote them seperately.

Infering a real-fake pair

> There are a fake coins and b real coins ($\mathtt{a}, \mathtt{b} > 0$). The real coins are all the same weight and also the counterfeit coins, but two types are different weight. Each time we compare only two coins have the same weight or not. In this case, assume a, b are known. Find the smallest number of comparison that we can infer a pair is real-fake. In this problem, we don't need to know which one is fake.

Definition 5. *If a scheme could infer a real-fake coin. We say this graph is a inferable$^{\pm}$ scheme for fake (IS$^{\pm}$). Otherwise it's not inferable$^{\pm}$.*

Optimal inferable scheme for fake (OIS$^{\pm}$) is IS$^{\pm}$ with minimum number of comparison. Deonte the number of edges of OIS$^{\pm}$ by $\tau^{\pm}(\mathtt{a}, \mathtt{b})$.

Note that in the foolproof problem. We can infer the pair on the scale which is unbalanced. But sometimes before the unbalance appears, we can infer a pair is real-fake. The following shows the answer is $\tau(\mathtt{a}, \mathtt{b}) - 1$.

Lemma 4. *OIS$^{\pm}$ = removing an arbitrary edgeThat is, $\tau^{\pm}(a, b) = \tau(a, b) - 1$*

Proof. We divide the proof of the equation into two inequations. For conciseness, we denote $\tau(a, b) - 1$ by τ and the answer of this problem as k_{min}.

For a optimal foolproof scheme G(with τ edges). Let $G' =$ (G remove an arbitrary edge e), then through F'. If G' has unbalance comparison, we conclude the one is unbalanced, otherwise we could conclude e is unbalanced. Either case shows G' is sufficient to infer a real-fake pair.

$$k_{min} \leq \tau - 1$$

If we could infer a pair e is real-fake through the optimal scheme F(with k_{min} comparisons). Then the graph $F + e$ is foolproof.

$$\tau \leq k_{min} + 1$$

From the above inequations, the lemma holds.

□

Two conditions of a testing scheme

When testing a testing scheme, there are two conditions.

Condition 1. there is an unbalanced edges

Condition 2. they are all balanced

For infering a $\pm$, we must ganrantee both conditions work. For Condition 1, we would simplily infer the unbalanced edges. For Condition 2, we would infer the edge must be unbalanced because adding this edge would form a foolproof scheme. For the following inference problems($-$,$+$,$+-$), we shall use the same logic.

Infering a fake coin

> There are $\mathtt{a}$ fake coins and $\mathtt{b}$ real coins $(\mathtt{a}, \mathtt{b} > 0)$. The real coins are all the same weight and also the counterfeit coins, but two types are different weight. Each time we compare only two coins have the same weight or not. In this case, assume $\mathtt{a}$, $\mathtt{b}$ are known. Find the smallest number of comparison that we can infer a coin is fake.

Definition 6. *If a scheme could infer a fake coin. We say this graph is a inferable$^-$ scheme for fake (IS$^-$). Otherwise it's not inferable$^-$.*

Optimal inferable scheme for fake (OIS$^-$) is IS$^-$ with minimum number of comparison. Deonte the number of edges of OIS$^-$ by $\tau^-(\mathtt{a}, \mathtt{b})$.

Note that $OFS(a, b)$ could find a real and a fake coin. Thus $\tau(\mathtt{a}, \mathtt{b})$ is an upper bound of $\tau^-(\mathtt{a}, \mathtt{b})$. A $IS^-(\mathtt{a}, \mathtt{b})$ better than $OFS(\mathtt{a}, \mathtt{b})$ must use another strategy, which is, though all comparisons result in balance we could still infer a '$-$'.

Before solving IS^-, we observe how a inferable$^-$ scheme works.

For a testing scheme, there are two conditions

If **Condition 1.** happens then we immediate infer the fake coin from the unbalanced. If **Condition 2.** happens, we must claim a fake coin, and this coin couldn't be a real one in any condition, otherwise this inference is wrong.

We present an optimal inferable$^-$ scheme as a integer partition for an example. Let (a,b)=(2,6), this scheme is [3,3,'1',1] .If **Condition 2**, we infer the '1' is the fake coin. To explain more, we introduce the following lemma.

Lemma 5. *If **Condition 2**, the scheme is inferable for a fake coin c_1 if and only if removing c_1's part, the remaining parts avoids $\mathtt{a}$. Physically, c_1 must not be a real coin with **Condition 2**.*

To find a inferable$^-$ better than OFS must solve **Condition 2**. Denote the part of the fake coin to be infered as $i (i \leq \mathtt{a})$. Beyound the remainning $\mathtt{a} + \mathtt{b} - i$ parts, (1)they must avoid $\mathtt{a}$, since if there's a subset sum $\mathtt{a}$, assign the subset '+', this claim is wrong (2)they must contain $\mathtt{a} - i$ or equivalently contain $\mathtt{b}$, since there must exist a condition such that the coin is fake and **Condition 2**.

Definition 7. *Same as previous problem, we define IS$^-$, OIS$^-$ and $\tau^-(a, b)$.*

Note that $Q(n, a, b) \geq \tau(a, n - a)$

For **Condition 2.**, $1 \leq i \leq \mathtt{a}$ for IS^-, thus we get the following lemma.

Lemma 6. $\tau^-(\mathtt{a}, \mathtt{b}) = min\{\tau(\mathtt{a}, \mathtt{b}), \{i - 1 + Q(\mathtt{a} + \mathtt{b} - i, \mathtt{a}, \mathtt{b}) | 1 \leq i \leq \mathtt{a}\}\}$

We are ready to solve OIS^- now.

Theorem 6. $OIS^-(\mathtt{a}, \mathtt{b}) = optimal\{OFS(a, b), \{$*a tree of i-1 nodes and* $OFS(a, b - i) | 1 \leq i \leq a\}\}$

$\tau^-(\mathtt{a}, \mathtt{b}) = min\{\tau(\mathtt{a}, \mathtt{b}), \{i - 1 + \tau(\mathtt{a}, \mathtt{b} - i) | 1 \leq i \leq \mathtt{a}\}\}$

Proof. We divide the proof into inferable$^-$ part and optimal part.
(inferable$^-$ Part:)
For $OFS(a,b)$ could infer a fake coin because only **Condition 1.** happens. For {a tree of i nodes and O $i)|1 \leq i \leq a\}\}$, if **Condition 2.**, we infer one coin of the i-node tree as fake coin.
(Optimal Part:) from lemma 5.2

$$\begin{aligned}\tau^-(a,b) &= min\{\tau(a,b), \{i-1+Q(a+b-i,a,b)|1 \leq i \leq a\}\}\\ &\geq min\{\tau(a,b), \{i-1+\tau(a,b-i)|1 \leq i \leq a\}\}\end{aligned}$$

is a optimal lower bound.
And $optimal\{OFS(a,b), \{\text{a tree of i nodes and } OFS(a,b-i)|1 \leq i \leq a\}\}$ reaches this lower bound.

□

Infering a real coin

There are a fake coins and b real coins ($\mathtt{a}, \mathtt{b} > 0$). The real coins are all the same weight and also the counterfeit coins, but two types are different weight. Each time we compare only two coins have the same weight or not. In this case, assume a, b are known. Find the smallest number of comparison that we can infer a coin is real.

Definition 8. *Same as previous problem, we define* IS^+, OIS^+ *and* $\tau^+(a,b)$.

OIS^+ is trivial. That is, comparing a coins in a group.

Theorem 7. $OIS^+(\mathtt{a},\mathtt{b})$ = *a tree of* $\mathtt{a}+1$ *nodes*

Proof. We divide the proof into inferable$^+$ part and optimal part.
(Inferable$^+$ Part:)
we omit this part since it's same as previous theorem.
(Optimal Part:)

$$\begin{aligned}\tau^+(a,b) &= min\{\tau(a,b), \{i-1+Q(a+b-i,b,a)|1 \leq i \leq b\}\}\\ &\geq min\{\tau(a,b), \{i-1+\tau(a-i,b)|1 \leq i \leq b\}\} && \text{by Theorem 2.2.}\\ &\geq min\{\lfloor\frac{a+b}{2}\rfloor, i-1+\frac{a-i+b}{2}\}\\ &\geq min\{a,a\} = a\end{aligned}$$

is a optimal lower bound.
a tree of $\mathtt{a}+1$ nodes reaches optimal lower bound. □

Infering a real coin and a fake coin

There are a fake coins and b real coins ($\mathtt{a}, \mathtt{b} > 0$). The real coins are all the same weight and also the counterfeit coins, but two types are different weight. Each time we compare only two coins have the same weight or not. In this case, assume a, b are known. Find the smallest number of comparison that we can infer a coin is real and another coin is fake.(e.g. a real-fake pair that we know which one is fake)

Definition 9. *Same as previous problem, we define* IS^{+-}, OIS^{+-} *and* $\tau^{+-}(a,b)$.

This problem seems the same as the foolproof problem, but not exactly. The difference is whether the real-fake pair need to be compared on the scale. For most cases, OFS is optimal for this problem but not all cases. That is, the answer is τ. Consider $(\mathtt{a},\mathtt{b}) = (3,8)$, there's a OFS, [2, 2, 2, 5]. An edge removed, the scheme [2, 2, 2, '4', '1'] is not only inferable$^{\pm}$ but also $^{+}-$('4' and '1').(see explaination in Figure) In this case, the answer is $\tau - 1$.

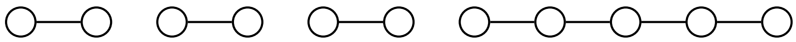

Figure 3: A foolproof scheme for $(\mathtt{a},\mathtt{b}) = (3,8)$

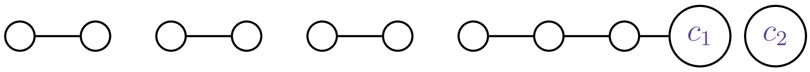

Figure 4: An IS^{+-} for $(\mathtt{a},\mathtt{b}) = (3,8)$. If there is an unbalance edge, we infer the fake and real coin of the pair. If edges are all balance, we can infer the coin c_1 is real and the coin c_2 is fake.

Before discussing how to find an IS^{+-}, we observe the range of $\tau^{+-}(a,b)$. Since $\tau(\mathtt{a},\mathtt{b})$ is sufficient to infer a fake coin and a real coin.

$$\tau^{+-} \leq \tau$$

Through we cound infer a fake coin and a real coin, add this edge into the OIS^{+-}, it's foolproof.

$$\tau \leq \tau^{+-} + 1$$

Combine the inequations

$$\tau - 1 \leq \tau^{+-} \leq \tau$$

Before solving IS^{+-}, we observe how this works.

If **Condition 1.** happens then the solution works. If **Condition 2.** happens, we infer a fake coin and a real coin. This pair is not in the IS^{+-}. Adding this pair(edge) into the IS^{+-}, the new graph is foolproof.

Lemma 7. *Any inferable*$^{+-}$ *scheme is either a foolproof scheme or a foolproof scheme with an edge removed.*

Theorem 8. *Any optimal inferable*$^{+-}$ *scheme is either a optimal foolproof scheme or an optimal foolproof scheme with an edge removed.*

Unlike infering a real-fake pair, removing an arbitrary edge from a *OFS* may not work. We enumerate the edge removed for each *OFS* then check whether it's inferable^{+-}. Checking a scheme's inferability^{+-} is similar to checking its inferability^{+}. Condition 1 is simple and the scheme to ganrantee Condition 2 don't happen is *FS*. We shall focus on Condition 2. If Condition 2 happen can we infer a $+-$? To explain more, we return to the case Figure 4, the reason Figure 4 works is from lemma 5. Because removing c_1's component, the multiset [2,2,2,1] avoids $\mathtt{b}$

6 Open questions

6.1 Cycle happens earlier

As in theorem 4.3 and 4.4, we proved $\rho(\mathtt{a}, \mathtt{b})$ would be in cycle for $\mathtt{b} > d(d+1)(\mathtt{a}-1) + (\mathtt{a} - (\mathtt{a} \bmod d)$. But observing ρtable in Results 3.2, ρ seems to get in cycle earlier. After computer comfirmation, we got results bellow.

a	b: $\rho(\mathtt{a}, \mathtt{b})$ start to cycle by theorem	b: $\rho(\mathtt{a}, \mathtt{b})$ start to cycle by computer comfirmation
1	1	1
2	13	1
3	15	5
4	40	10
5	29	9
6	105	29
7	43	13
8	91	22
9	57	17
10	118	28

Since the algorithm cost exponential time, we could only comfirm this in small case.

But if it get in cycle earlier in general case, we can use a better(smaller) K algorithm 4.1 and run faster.

6.2 Avoidance integer partition

As in algorithm 3.1,4.1, to find an optimal solution require enumerating partitions of $\mathtt{a} + \mathtt{b}$ avoiding $\mathtt{a}$. In the algorithms, we enumerating all partitions of $\mathtt{a} + \mathtt{b}$ and check if it avoids $\mathtt{a}$. So the time complexity is analyzed as the (number of partitions) times checking avoidance. If exist a way to enumerate avoidance and bound it tighter, the time complexity would be better.

References

[1] Howard D. Grossman. The Twelve-Coin Problem, *Scripta Mathematica* XI, pages 360–361, 1945.

[2] Richard K. Guy and Richard J. Nowakowski. Coin-Weighing Problems, *The American Mathematical Monthly*, 102(2):164–167, 1995.

[3] Lorenz Halbeisen and Norbert Hungerbühler. The General Counterfeit Coin Problem, *Discrete Mathematics*, 147(1–3):139–150, 1995.