CSIT113
Problem Solving

Workshop - Week 8

Fake coin 1

- There are eight identical-looking coins; one of these coins is counterfeit is known to be lighter than the genuine coins.
- What is the minimum number of weighings needed to identify the fake coin with a two-pan balance scale without weights?

Solution

- The answer is two weighings.
- Select from the given coins two groups of three coins each and put them on the opposite cups of the scale. If they weigh the same, the fake is among the other two coins, and weighing these two coins will identify the lighter fake.
- If the first weighing does not yield a balance, the lighter fake is among the three lighter coins.
- Take any two of them and put them on the opposite cups of the scale. If they weigh the same, it is the third coin in the lighter group that is fake; if they do not weigh the same, the lighter one is the fake.
- Since the problem cannot be solved in one weighing, the above algorithm requiring just two weighings is optimal.

Another solution

- The problem has an alternative solution in which the second weighing does not depend on the results of the first one.
- Label the coins by the letters A, B, C, D, E, F, G, H.
- On the first weighing, weigh A, B, C against F, G, H.
- On the second weighing, weigh A, D, F against C, E, H.
- If ABC = FGH (the first weighing results in a balance), all these six coins are genuine, and therefore the second weighing is equivalent to weighing D against E.
- If ABC < FGH, only A, B, and C may still be fake. Therefore if on the second weighing ADF = CEH, B is the fake; if ADF<CEH, A is the fake; and if ADF > CEH, C is the fake.
- The case of ABC>FGH is symmetric to the case just discussed.

Fake coin 2

- You have $n > 1$ identical-looking coins: $n - 1$ of them are genuine with a known weight g, and one of them - of an unknown weight different from g - is counterfeit.
- Design an algorithm that determines the fake in the minimum number of weighings on a spring scale.
- Assume that then spring scale indicates the exact weight of the coins being weighed.

Solution

- The minimum number of weighings needed to guarantee identification of the fake coin is $\lceil log_2 n \rceil$.
- Consider an arbitrary subset *S* of *m* ≥ 1 coins selected from the *n* coins given.
- If its total weight *W* is equal to *gm,* all the coins in *S* are genuine; otherwise, one of the coins in *S* is fake.
- Hence we can proceed searching for the fake among the coins not in *S* in the former case and among the coins in *S* in the latter case.

- If *S* contains half (or nearly half) the coins given, after one weighing we can proceed searching for the fake in a set half the size.
- We will have to repeat this halving and weighing operation $\lceil log_2 n \rceil$ times before *n*, the size of the original set, can be guaranteed to be reduced to 1.
- Formally, the number of weighings in the worst case is defined by the recurrence $W(n) = W(\lceil n/2 \rceil) + 1$ for $n > 1$, $W(1) = 0$.
- Hence the solution is $W(n) = \lceil log_2 n \rceil$

Fibonacci’s Rabbits Problem

- A man put a pair of rabbits in a place surrounded on all sides by a wall.
- The initial pair of rabbits (male and female) are newborn.
- All rabbit pairs are not fertile during their first month of life but give birth to one new male/female pair at the end of the second month and every month thereafter.
- How many pairs of rabbits will be there in a year?

Solution

- Let $R(n)$ be the number of rabbit pairs at the end of month n.
- Clearly, $R(0) = 1$ and $R(1) = 1$
- For every $n > 1$, the number of rabbit pairs, $R(n)$, is equal to the number of pairs at the end of month $n - 1$, $R(n - 1)$, plus the number of rabbit pairs born at the end of month n
- According to the problem's assumptions, the number of newborns is equal to $R(n - 2)$, the number of rabbit pairs at the end of month $n - 2$

- Thus, we have the recurrence relation

$$R(n) = R(n - 1) + R(n - 2) \text{ for } n > 1,\ R(0) = 1,\ R(1) = 1.$$

- The following table gives the values of the first 13 terms of the sequence, called the *Fibonacci numbers*, defined by this recurrence relation:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$R(n)$	1	1	2	3	5	8	13	21	34	55	89	144	233

- Hence, after 12 months, there will be 233 pairs of rabbits.

Searching a Sorted Table

- One hundred different numbers are written on 100 cards, one number per card.
- The cards are arranged in 10 rows and 10 columns, in increasing order in each row (left to right) and each column (top down).
- All the cards are turned faced down so that you cannot see the numbers written on them.
- Can you devise an algorithm to determine whether a given number is written on one of the cards by turning up less than 20 cards?

Solution

- Let us start by turning up the card at the upper right corner of the array and compare its number with the number we are searching for.
- If the numbers are the same, the problem is solved
- If the search number is smaller than the number on the card, the search number cannot be in the last column, and we can move left to the card in the preceding column.
- If the search number is larger than the number on the card, the search number cannot be in the first row, and we can move down to the card in the next row.
- Repeating this operation until either a search number is found or the search leads outside the array solves the problem.

- The algorithm's sequence of turned up cards forms a zigzag line made up of segments going left or down from the upper right corner to some card in the array.
- The longest such line ends at the lower left corner, turning up the total of 19 cards.
- No such line can be longer, because it cannot have more than nine horizontal and nine vertical segments.