

## 4-5 Chip testing

Professor Diogenes has  $n$  supposedly identical integrated-circuit chips that in principle are capable of testing each other. The professor's test jig accomodates two chips at a time. When the jig is loaded, each chip tests the other and reports whether it is good or bad. A good chip always reports accurately whether the other chip is good or bad, but the professor cannot trust the answer of a bad chip. Thus, the four possible outcomes of a test are as follows:

Chip $A$ says	Chip $B$ says	Conclusion
$B$ is good	$A$ is good	both are good, or both are bad
$B$ is good	$A$ is bad	at least one is bad
$B$ is bad	$A$ is good	at least one is bad
$B$ is bad	$A$ is bad	at least one is bad

- a. Show that if more than  $n/2$  chips are bad, the professor cannot necessarily determine which chips are good using any strategy based on this kind of pairwise test. Assume that the bad chips can conspire to fool the professor.
- b. Consider the problem of finding a single good chip from among  $n$  chips, assuming that more than  $n/2$  of the chips are good. Show that  $\lfloor n/2 \rfloor$  pairwise tests are sufficient to reduce the problem to one of nearly half the size.
- c. Show that the good chips can be identified with  $\Theta(n)$  pairwise tests, assuming that more than  $n/2$  chips are good. Give and solve the recurrence that describes the number of tests.

a. Lets say that there are  $g < n/2$  good chips and  $n - g$  bad chips.

From this assumption, we can always find a set of good chips  $G$  and a set of bad chips  $B$  of equal size  $g$  since  $n - g \geq g$ .

Now, assume that chips in  $B$  always conspire to fool the professor in the following:

"for any test made by the professor, chips in  $B$  declare chips in  $B$  as 'good' and chips in  $G$  as 'bad'."

Since the chips in  $G$  always report correct answers thus there exists symmetric behaviors, it is not possible to distinguish bad chips from good ones.

b.

Generalize the original problem to: "Assume there are no less good chips than bad chips."

**Algorithm:**

1. Pairwise test them, and leave the last one alone if the number of chips is odd.
  - If the report says at least one of them is bad, throw both chips away;
  - otherwise, throw one away from each pair.
2. Recursively find one good chip among the remaining chips. The recursion ends when the number of remaining chips is 1 or 2.
  - If there is only 1 chip left, then it is the good chip we desire.
  - If there are 2 chips left, we make a pairwise test between them. If the report says both are good, we can conclude that both are good chips. Otherwise, one is good and the other is bad and we throw both away. The chip we left alone at step 1 is a good chip.

**Explanation:**

1. If the number of chips is odd, from assumption we know the number of good chips must be greater than the number of bad chips. We randomly leave one chip alone from the chips, in which good chips are not less than bad chips.
2. Chip pairs that do not say each other is good either have one bad chip or have two bad chips, throwing them away doesn't change the fact that good chips are not less than bad chips.
3. The remaining chip pairs are either both good chips or bad chips, after throwing one chip away in every those pairs, we have reduced the size of the problem to at most half of the original problem size.
4. If the number of good chips is  $n$  ( $n > 1$ ) more than that of bad chips, we just throw away the chip we left alone when the number of chips is odd. In this case, the number of good chips is at least one more than that of bad chips, and we can eventually find a good chip as our algorithm claims.
5. If the number of good chips is exactly one more than that of bad chips, there are 2 cases.
  - We left alone the good chip, and remaining chips are one half good and one half bad. In this case, all the chips will be thrown away eventually. And the chip left alone is the one we desire.
  - We left alone the bad chip, there are more good chips than bad chips in the remaining chips. In this case, we can recursively find a good chip in the remaining chips and the left bad chip will be thrown away at the end.

c. As the solution provided in (b), we can find one good chip in

$$T(n) \leq T(\lceil n/2 \rceil) + \lfloor n/2 \rfloor.$$

By the master theorem, we have  $T(n) = O(n)$ . After finding a good chip, we can identify all good chips with that good chip we just found in  $n - 1$  tests, so the total number of tests is

$$O(n) + n - 1 = \Theta(n).$$