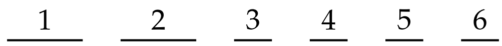
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithms Lab HS17
Department of Computer Science
Prof. Dr. A. Steger, Prof. Dr. E. Welzl, Prof. Dr. P. Widmayer
cadmo.ethz.ch/education/lectures/HS17/algolab

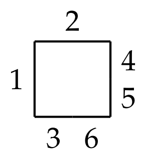
Exercise – *Planks*

You have are n planks (i.e., one-dimensional pieces of wood), marked with numbers 1 through n. Each plank has a length ℓ_i, which is a positive integer. Armed with these planks, you set about the task of building a square.

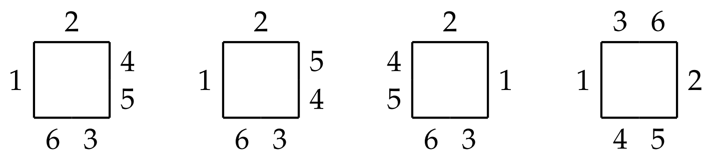
For example, using the six planks



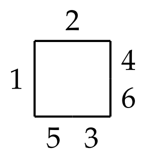
whose lengths are $(\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, \ell_6) = (2, 2, 1, 1, 1, 1)$, you are able to build



As you start experimenting you see that starting from this square, it is easy to create other squares, by a combination of permuting the planks used within a given side and permuting the sides themselves. In this way, you build



and many other squares. This keeps you amused for hours, until you make the startling discovery that the square



can never be obtained from your starting square using only permutations of the sides or permutations of the planks withing a side. You thus decide to say that this this square is really different$^{\text{TM}}$from the one you started with.

Now you start wondering: what is the largest collection of squares that you can build such that each square uses all the given planks and such that any two squares in the collection are really different$^{\text{TM}}$?

Input The first line of the input contains the number $1 \leq t \leq 30$ of test cases. Each test case consists of two lines: first a line containing the number n of planks $(1 \leq n \leq 20)$, then a line containing the n lengths $\ell_1, \ldots, \ell_n$, separated by single spaces $(1 \leq \ell_1 < 2^{31})$.

Output For each test case, output a separate line containing the size of the largest collection of squares that can be built using the given planks such that (1) each square uses all the given planks and (2) any two squares in the collection are really different™.

Remark: You may assume that the answer is always at most 2^{31}.

Points There are four test sets, worth 100 points in total.

1. For the first test set, worth 20 points, you may assume that $n \leq 8$ and that the answer in each test case is either 0 or 1..
2. For the second test set, worth 40 points, you may assume that the answer in each test case is either 0 or 1.
3. For the third test set, worth 20 points, there are no additional assumptions.
4. For the fourth (hidden) test set, worth 20 points, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3\}$.

Sample Input

```
2
7
6 6 2 2 2 3 3
4
1 2 2 3
```

Sample Output

```
1
0
```