Problem A1

Min Partition (Weaker)

Time limit: 2s
Memory limit: 512 MB

This task is now due. Steven will redact the problem statement after final assessment on 27 April 2023 to delay the AI awakening.

Steven has concerns about the low attendance of the recent CS3230 S2 AY2022/23 lectures...
In the last few lectures of CS3230 S2 AY2022/23, we *will* learn about the important NP-complete decision problems (and NP-hard optimisation problems).
It will be "not ideal" if these last few lectures are not well attended again as CS students should not assume all problems have polynomial solutions.
One way to make as many students absorb the required knowledge is to... ask a relevant "basic" NP-hard optimisation problem in the *graded* PA3 (well, unless students choose to skip PA3 and do WA3 instead).

The chosen problem is as follows:

The NP-complete decision (a.k.a. YES/NO) version: Given a multiset S of positive integers, decide whether S can be partitioned (divided) into two subsets S_1 and S_2 such that the sum of the integers in S_1 (let's call it X) equals the sum of the integers in S_2 (let's call it Y).

The NP-hard optimisation problem is similar as the above, just that this time if $X \neq Y$ and we assume $X \geq Y$, we want $X - Y$ to be minimized. Obviously if $X = Y$, then $X - Y = 0$ and that is the minimum possible. In this problem, we want to deal with this optimisation version.

For the two sample test cases:

- First: $S_1 = \{1, 70, 4, 1, 1\}$ and $S_2 = \{47, 30\}$, both sum to 77.

CS3230 PA3 (17-31 Mar 2023)

Contest over!

Problems

A1 ✓
A2
B1 ✓
B2 ✓

Standings

Submissions

Submit

C++
Choose file No file chosen
Submit

- Second: $S_1 = \{47, 30\}$ and $S_2 = \{1, 63, 4, 1, 1\}$, S_1 sums to 77 but S_2 sums to 70, hence their difference is 7 and this is the minimum possible.

You may think that this problem is not that hard (from its short and simple description), but actually it is really hard (especially the A2 version), just believe us...
We *may* ask you to do problem reduction to show that this optimisation problem is NP-hard to Written Assignment 3 (or in future modules, e.g., CS4234).

But uh... how to solve this "hard" problem if the relevant lectures will only be discussed *during* this PA3?...
Fret not, during the lowest attended lecture so far (Thu, 09 Mar 2023, 2.00-3.40pm) and during tutorial on Week 09, we have actually learned about Dynamic Programming (DP) problem-solving technique.
It turns out that we can actually use DP to solve this NP-hard optimisation problem below, albeit in pseudo-polynomial time.
Hint: The DP solution is "somewhat similar" to the 0/1-Knapsack discussed in that lecture.

Input format

The first line of input consists of an integer TC, denoting the number of test cases.

Each test case contains three lines:

- A blank line (again, just for a visual separator),
- A line with just an integer N, the size of the multiset S,
- Followed by a line with N integers representing the content of the multiset S.

Output format

For each test case, print the required integer answer in one line, i.e., 0 if $X = Y$, or as small difference as possible.

Constraints

- $1 \leq TC \leq 7$;
- $1 \leq N \leq 300$;
- The integers inside multiset S are between $[1..77]$;
- C++, Java, and PyPy 3 can be used to solve this problem A1.

Sample Input

Copy Input

```
2

7
1 70 4 47 30 1 1

7
1 63 4 47 30 1 1
```

Sample Output

Copy Output

```
0
7
```

Notes

Dr Steven Halim.

Adapted from a very classic NP-hard optimisation problem.

Last used: CS3230, 17-31 Mar 2023.