

Algorithms Lab

Exercise – Burning Coins from Two Sides

A friend of yours who recently sold his internet startup for several million dollars wants to share parts of his money with you. The catch is that he will not just give it to you but that he wants to play a game that will determine how much you actually get. The game works as described below.

There are n coins lined up in a row. Each coin has a specific known value v_i . The game is played in turns and you are the one to start. Every time it is your turn you can either take the first or last coin in the row. Once you took the coin of your choice your friend will do the same (also either take the now first or last coin in the row), and this procedure will be iterated until someone has taken the last coin. The money you get at the end is the sum of the values of the coins you have collected while playing this game.

For a given row of coins with corresponding values v_1, v_2, \dots, v_n , you want to know what is the largest amount of money you are *guaranteed* to win if you manage to play optimally, assuming you start the game. (Independent of the strategy of your friend.)

Input The first line of the input contains the number $t \leq 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that contains one integer n denoting the number of coins used for the game, such that $1 \leq n \leq 2500$.
- The following line contains n integers $v_0 \dots v_{n-1}$, separated by a space, where v_i denotes the value of the i th coin, and such that $1 \leq v_i \leq 1000$.

Output For each test case output a single line with one integer denoting the largest amount you are *guaranteed* to win if you start the game.

Points There are three groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 30 points, you may assume that $n \leq 10$.
2. For the second group of test sets, worth 40 points, you may assume that $n \leq 500$.
3. For the third group of test sets, worth 30 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
3
3 1 2
4
1 4 9 4
```

Sample Output

```
4
10
```