Seating Arrangements

There are n guests attending a dinner party, numbered from 1 to n. The ith guest has a height of `arr[i-1]` inches.

The guests will sit down at a circular table which has n seats, numbered from 1 to n in clockwise order around the table. As the host, you will choose how to arrange the guests, one per seat. Note that there are n! possible permutations of seat assignments.

Once the guests have sat down, the awkwardness between a pair of guests sitting in adjacent seats is defined as the absolute difference between their two heights. Note that, because the table is circular, seats 1 and n are considered to be adjacent to one another, and that there are therefore n pairs of adjacent guests.

The overall awkwardness of the seating arrangement is then defined as the maximum awkwardness of any pair of adjacent guests. Determine the minimum possible overall awkwardness of any seating arrangement.

Signature

```
int minOverallAwkwardness(int[] arr)
```

Input

n is in the range [3, 1000].
Each height `arr[i]` is in the range [1, 1000].

Output

Return the minimum achievable overall awkwardness of any seating arrangement.

Example

```
n = 4
arr = [5, 10, 6, 8]
output = 4
```

If the guests sit down in the permutation [3, 1, 4, 2] in clockwise order around the table (having heights [6, 5, 8, 10], in that order), then the four awkwardnesses between pairs of adjacent guests will be |6-5| = 1, |5-8| = 3, |8-10| = 2, and |10-6| = 4, yielding an overall awkwardness of 4. It's impossible to achieve a smaller overall awkwardness.