Problem: Log Jumping

Problem Description

We want to arrange N logs in a circle for a playing ground shown in Figure 1. Kids are jumping one log to the next log circularly. We are concerned to arrange the logs to help kids by minimizing the height difference between two adjacent logs.

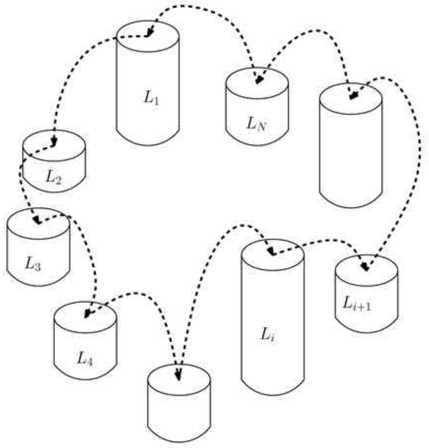

Figure 1. N logs are arranged in a circle

The *difficulty* of log jumping depends on the maximum difference of heights among all pairs of two adjacent logs. Assume that we are given 5 logs with the heights {2, 4, 5, 7, 9}. Note that the last log with height 5 is immediate adjacent to the first log with height 2 in the arrangement [2, 9, 7, 4, 5]. The maximum height difference of the arrangement [2, 9, 7, 4, 5] is $|2-9|=7$ We can make a better arrangement as [2, 5, 9, 7, 4] where the maximum height difference is $|5-9|=4$. This arrangement [2, 5, 9, 7, 4] is optimal since there is no arrangement with the maximum height difference less than 4.

Input

The name of input file is 'jump.inp.' The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer, $N(5 \leq N \leq 10{,}000)$, where N is the number of logs. In the next line, the heights of N logs, L_i are given as a sequence of integers. $(1 \leq L_i \leq 100{,}000)$

Output

The name of output file is 'jump.out.' Print exactly one line for each test case. The line should contain an integer representing the maximum height difference of an optimal log arrangement. The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input

3
7
13 10 12 11 10 11 12
5
2 4 5 7 9
8
6 6 6 6 6 6 6 6
	1
	4
	0

Constraints: The file name should be 'jump.{c, cpp, java}.'