Two Array Problem

In this problem you operate on two arrays of N integers. We will call them the 0^{th} and the 1^{st} respectively.
Your goal is just to maintain them under the modification operations, such as:

- 1 $id\ l\ r$: Reverse the subarray of the id^{th} array, starting at the l^{th} number, ending at the r^{th} number, inclusively;
- 2 $id\ l_1\ r_1\ l_2\ r_2$: Swap two consecutive fragments of the id^{th} array, the first is from the l_1^{th} number to the r_1^{th}, the second is from the l_2^{th} number to the r_2^{th};
- 3 $l\ r$: Swap the piece that starts at the l^{th} number and end at the r^{th} one between the 0^{th} and the 1^{st} array;
- 4 $l\ r$: We consider only the piece from the l^{th} number to the r^{th} one. The numbers in the 0^{th} array are X-coordinates of some set of points and the numbers in the 1^{st} array are Y-coordinates of them. For the obtained set of points we would like to place such a circle on a plane that would contain all the points in it and would have the minimal radius. Find this minimal radius.

Input Format
The first line of input contains two space separated integers N and M denoting the number of integers in arrays and the number of queries respectively.
The second line contains N space separated integers: the initial elements of the 0^{th} array.
The third line contains N space separated integers: the initial elements of the 1^{th} array.
Then there are M lines containing queries in the format listed above.

Output Format
For each type-**4** query output the sought minimal radius with **exactly** two symbols after the decimal point precision.

Constraints
$1 \leq N, M \leq 10^5$
All the numbers in arrays are non-negative and don't exceed 10^6.
The sum of $R - L$ over the type-**4** queries won't exceed 10^6.
In the query of the type 2, $1 \leq l_1 \leq r_1 < l_2 \leq r_2 \leq N$.
In the queries of the types 1, 3, 4, $1 \leq l \leq r \leq N$; $0 \leq id < 2$.

Sample Input

```
10 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
3 2 6
1 0 9 9
4 6 9
2 0 2 7 9 9
1 0 3 6
2 1 2 3 4 5
1 1 7 10
2 1 8 8 9 10
4 6 9
2 0 2 2 4 6
```

Example Output

2.12
2.50