Problem 4 - COOL Server (100 pts)

Problem Description

Serving thousands of students, NTU COOL has N machines to execute programs. In order to check whether the machines run normally, the administrators may choose a block of machines (with consecutive ID) and reboot them **one by one**. At the beginning, assume that the i-th machine needs a_i second(s) to reboot. Long down time causes inconvenience, so the administrators must know exactly how much time the machines need to reboot.

It seems like an easy thing. However, during the semester, the NTU COOL team may change the positions of the machines, buy new ones and (or) discard a broken one. Moreover, they may update a block of machines (with consecutive IDs) to improve the user experience. With these operation, estimating the required time to reboot machines becomes difficult. Please help them to compute the required time to reboot based on the machine maintenance records.

Input Format

The first line contains two integers N, Q, representing the number of machines, and the number of records, respectively. Let the 1-st machine be the leftmost one and the N-th machine be the rightmost one.

The next line contains N positive integers $a_1, a_2, \ldots, a_N$, representing the required time for the i-th machine to reboot, in second.

Each of the next Q lines is given in one of the following formats:

- `1 p k`: A new machine which needs k seconds to reboot is placed between the p-th machine and the $(p+1)$-th machine. If $p = 0$, the new machine is placed to the left of the originally 1-st machine. If $p =$ the number of machines, the new machine is placed to the right of the originally rightmost machine.
- `2 p`: The p-th machine will be retired and discarded.
- `3 l r`: The order of the machines from the l-th to the r-th is reversed, which would swap the l-th machine and the r-th machine, swap the $(l+1)$-th machine and the $(r-1)$-th machine, and so on.
- `4 l r x y`: Swap the block of machines between the l-th and the r-th and the block of machines between the x-th and the y-th. Note that the number of machines in these two specified blocks are **NOT necessarily the same**.

- `5 l r k`: For any machine between the l-th and the r-th, if more than k second(s) is required to reboot, update it so that only k second(s) is needed to reboot.
- `6 l r`: Output a line with an integer representing the amount of time required to reboot the machines from the l-th to the r-th one by one, in second.

Output Format

For each query, output a line with an integer representing the amount of time required to reboot the machines from the l-th to the r-th one by one, in second.

Constraint

- $1 \leq N, Q \leq 10^5$
- all the indices in the operation (p, l, r, x, y) are legal and form a proper range
- for operation `3,4,5,6`, $l \leq r, x \leq y$
- for operation `4`, $[l, r]$ and $[x, y]$ do not overlap
- $1 \leq a_i, k \leq 10^9$

Subtasks

Subtask 1 (5 pts)

- $N, Q \leq 1000$
- Only operation `6`

Subtask 2 (5 pts)

- Only operation `6`

Subtask 3 (5 pts)

- Only operation `1,2,3,4,5`

Subtask 4 (5 pts)

- Only operation `1,6`

Subtask 5 (5 pts)

- Only operation `2,6`

Subtask 6 (5 pts)

- Only operation `1,2,6`

Subtask 7 (10 pts)

- Only operation `3,6`

Subtask 8 (10 pts)

- Only operation `4,6`

Subtask 9 (10 pts)

- Only operation `5,6`

Subtask 10 (20 pts)

- Only operation `3,4,5,6`

Subtask 11 (20 pts)

- No other constraints!

Sample Cases

Sample Input 1

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

Sample Output 1

```
16
6
4
23
```

Sample Input 2

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

Sample Output 2

```
18
3
2
16
```

Hints

- The Problem Description is actually a noise. Getting rid of it may be useful.
- One good strategy is to split the subtasks and use different ways to solve them, and then merge all the method together to solve the whole problem.