# Team Building

You aim to assemble a team of $N$ programmers. You've already scouted them and assessed that the skill level of the $i$-th individual ($1 \leq i \leq N$) is represented by the nonnegative integer $s[i]$. You've realized that what truly matters is the order in which you hire them.

Each programmer is characterized by two additional integer values: workrate and motivation, both of which are $0$ upon their arrival but can increase after hiring new team members. When a new programmer is hired, the following events occur in the given order:

- The new programmer joins the team with workrate and motivation initialized to $0$.
- The workrate of each other previously hired programmer is increased by their own motivation value.
- The motivation of each other previously hired programmer is increased by the skill level of the new hire.

The strength of the team is determined afterwards by the sum of the workrates of all the team members. Your objective is to calculate the maximum attainable team strength by optimizing the order of hiring.

For example, if you hire programmers with skill levels $(0, 2, 2, 3)$ in this order, the hiring process will affect their values as follows:

| Event | Workrates | Motivations |
|---|---|---|
| Hiring with skill 0 | **0** | **0** |
| Hiring with skill 2 | 0 **0** | 0 **0** |
| Workrates update | **0** 0 | 0 0 |
| Motivations update | 0 0 | **2** 0 |
| Hiring with skill 2 | 0 0 **0** | 2 0 **0** |
| Workrates update | **2 0** 0 | 2 0 0 |
| Motivations update | 2 0 0 | **4 2** 0 |
| Hiring with skill 3 | 2 0 0 **0** | 4 2 0 **0** |
| Workrates update | **6 2 0** 0 | 4 2 0 0 |
| Motivations update | 6 2 0 0 | **7 5 3** 0 |

The team strength will be calculated as $6+2+0+0=8$. However, if you hire programmers in better order $(2,2,3,0)$, you will achieve a team strength of $7+3+0+0=10$.

| New hire skill | Workrates | Motivations |
|---|---|---|
| 2 | 0 | 0 |
| 2 | 0 0 | 2 0 |
| 3 | 2 0 0 | 5 3 0 |
| 0 | 7 3 0 0 | 5 3 0 0 |

Furthermore, over the course of the upcoming $Q$ days, you will receive notifications about changes in the skill level assessments of certain programmers. After day $i$, the skill level of programmer $x[i]$ will be updated to $y[i]$ (which may match the previous value). This updated skill value will be used in the following days, until it potentially gets updated again.

After each day, starting from today, your goal is to determine the maximum achievable team strength by hiring all $N$ programmers, taking into account the assessed skill levels at that particular moment.

## Input Format

The first line contains two integers: $N$ and $Q$.

The second line contains integers: $s[1]$, $s[2]$, ... , $s[N]$.

Subsequently, there are $Q$ lines, the $i$-th of which contains two integers: $x[i]$ and $y[i]$.

## Output Format

Print $Q+1$ lines, each containing a single integer. These integers represent the maximum potential team strength after each day, in chronological order.

## Example

| Standard input | Standard output |
|---|---|
| 4 2 | 10 |
| 2 0 2 3 | 14 |
| 2 4 | 12 |
| 4 0 | |

The solution for the initial state is illustrated above. After the first day, the skill levels will be updated to $(2, 4, 2, 3)$ and the maximum attainable team strength becomes $14$, and after the second day, they will be further adjusted to $(2, 4, 2, 0)$.

## Constraints

- $2 \leq N \leq 50\ 000$
- $1 \leq Q \leq 100\ 000$
- $0 \leq s[i] \leq 100\ 000$ for each $1 \leq i \leq N$.
- $1 \leq x[i] \leq N$ for each $1 \leq i \leq Q$.
- $0 \leq y[i] \leq 100\ 000$ for each $1 \leq i \leq Q$.

## Subtasks

1. (11 points) $N \leq 7$; $Q \leq 100$
2. (19 points) $N, Q \leq 500$
3. (15 points) $Q \leq 10$
4. (6 points) The skill levels never exceed $1$.
5. (9 points) The skill levels never exceed $500$.
6. (12 points) $x[i] = 1$ for each $1 \leq i \leq Q$.
7. (10 points) Each update will change the skill level by at most $1$.
8. (18 points) No additional constraints.