

Team Building

Ваша цель - собрать команду из N программистов. Вы уже провели наблюдение и оценили уровень навыков i -го программиста ($1 \leq i \leq N$), представленный неотрицательным целым числом $s[i]$. Вы осознали, что очень важно в каком порядке вы будете их нанимать.

Каждый программист характеризуется двумя дополнительными целочисленными значениями: эффективностью и мотивацией, оба изначально равны нулю. Однако они могут увеличиваться после принятия новых членов в команду. Когда новый программист присоединяется к команде, происходит следующее в указанной последовательности:

- Новый программист присоединяется с начальной эффективностью и мотивацией, равными 0.
- Эффективность каждого ранее принятого программиста увеличивается на его текущую мотивацию.
- Мотивация каждого ранее принятого программиста увеличивается на уровень навыков нового программиста.

После этого мощность команды определяется как сумма эффективностей всех членов команды. Вам нужно посчитать максимально достижимую мощность команды, оптимизируя порядок найма(принятия на работу).

Например, если вы нанимаете программистов с уровнями навыков $(0, 2, 2, 3)$ в данном порядке, процесс найма повлияет на их значения следующим образом:

События	Эффективности	Мотивации
Найм с уровнем навыков 0	0	0
Найм с уровнем навыков 2	0 0	0 0
Обновление эффективностей	0 0	0 0
Обновление мотивации	0 0	2 0
Найм с уровнем навыков 2	0 0 0	2 0 0
Обновление эффективностей	2 0 0	2 0 0
Обновление мотивации	2 0 0	4 2 0
Найм с уровнем навыков 3	2 0 0 0	4 2 0 0
Обновление эффективностей	6 2 0 0	4 2 0 0
Обновление мотивации	6 2 0 0	7 5 3 0

Мощность команды будет рассчитана как $6 + 2 + 0 + 0 = 8$. Однако, если вы нанимаете программистов в более выгодном порядке $(2, 2, 3, 0)$, вы достигнете мощности команды в $7 + 3 + 0 + 0 = 10$.

Новый уровень навыков	Эффективности	Мотивации
2	0	0
2	0 0	2 0
3	2 0 0	5 3 0
0	7 3 0 0	5 3 0 0

Более того, в течение предстоящих Q дней вы будете получать уведомления о изменениях в оценках уровней навыков некоторых программистов. После дня i уровень навыков программиста $x[i]$ будет обновлен до $y[i]$ (что может совпадать с предыдущим значением). Это обновленное значение навыков будет использоваться в последующие дни, пока оно, возможно, снова не будет обновлено.

После каждого дня, начиная с сегодняшнего, вашей целью является определение максимально достижимой мощности команды при найме всех N программистов, учитывая оценки уровней навыков на этот конкретный момент.

Input Format

Первая строка содержит два целых числа: N и Q .

Вторая строка содержит целочисленные значения: $s[1], s[2], \dots, s[N]$.

Затем следует Q строк, где i -я строка содержит два целых числа: $x[i]$ и $y[i]$.

Output Format

Выведите $Q + 1$ строку, в каждой из которых содержится одно целое число. Эти числа представляют максимально возможную мощность команды после каждого дня в хронологическом порядке.

Example

Стандартный ввод	Стандартный вывод
4 2	10
2 0 2 3	14
2 4	12
4 0	

Решение для начального состояния представлено выше. После первого дня уровни навыков будут обновлены до $(2, 4, 2, 3)$, и максимальная достижимая мощность команды станет равной 14. А после второго дня они будут дополнительно изменены до $(2, 4, 2, 0)$.

Constraints

- $2 \leq N \leq 50\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq s[i] \leq 100\,000$ для каждого $1 \leq i \leq N$.
- $1 \leq x[i] \leq N$ для каждого $1 \leq i \leq Q$.
- $0 \leq y[i] \leq 100\,000$ для каждого $1 \leq i \leq Q$.

Subtasks

1. (11 баллов) $N \leq 7$; $Q \leq 100$
2. (19 баллов) $N, Q \leq 500$
3. (15 баллов) $Q \leq 10$
4. (6 баллов) Уровни навыков никогда не превышают 1.
5. (9 баллов) Уровни навыков никогда не превышают 500.
6. (12 баллов) $x[i] = 1$ для каждого $1 \leq i \leq Q$.
7. (10 баллов) Каждое обновление изменит уровень навыков максимум на 1.
8. (18 баллов) Нет дополнительных ограничений.