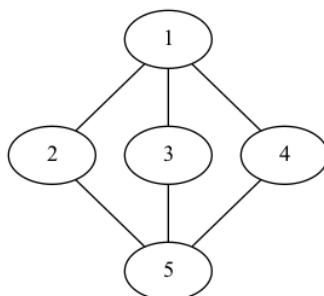


Problem Hide and seek

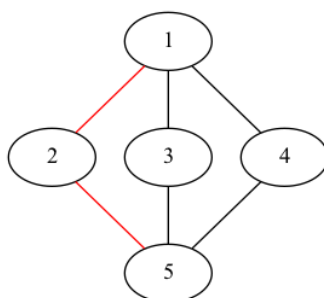
C++ header `hideandseek.h`

Танака та Лулу трошки нервують через cook-off змагання (тим більш, це найважливіше змагання в усьому Info(1)cup Королівстві). Тому, для того, щоб трошки відпочити, вони вирішили зіграти в хованки. Оскільки вони люблять змагатись, вони будуть грати в хованки на найбільшій поверхні, на якій можливо: по всьому Info(1)cup Королівству. Info(1)cup Королівство може бути розглянуте як мережа з n міст та m двосторонніх доріг. Два міста можуть бути з'єднані *щонайбільше однією дорогою* та *ніяка дорога не з'єднує місто саме з собою*. Гарантується, що можливо дібратись від будь-якого міста до будь-якого іншого. Лулу та Танака будуть ховатись в містах.

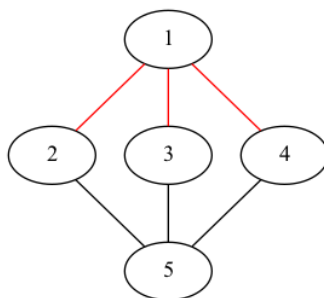
Лулу та Танака мають k друзів, які будуть намагатись перешкодити їм знайти один одного. Королівство має особливу властивість: поки Лулу та Танака ховаються в різних містах, кожен з k друзів може заблокувати дорогу, так, що не буде шляху між Лулу та Танакою. Друзям дозволяється не робити нічого (тобто не блокувати дорогу). Наприклад, уявімо, що королівство має 5 міст з дорогами між містами (1, 2), (1, 3), (1, 4), (2, 5), (3, 5), (4, 5). Зображення королівства нижче:



В цьому випадку, можливо мати $k = 3$ друзів. Якщо Лулу чи Танака ховаються в містах 2, 3 або 4, тоді їх друзі можуть заблокувати дві дороги, що приєднані до цих міст. Наприклад, якщо один ховається в 2, то заблокувавши дороги позначені червоним на малюнку нижче, їх друзі можуть зробити так, що не буде шляху між Лулу та Танакі:



З іншої сторони, якщо Лулу захирається в місті 1, а Танака в місті 5, тоді друзі можуть заблокувати дороги (1, 2), (1, 3), (1, 4), як на малюнку нижче:



Аналогічно, вони можуть мати $k = 3, k = 4$ або навіть $k = 5$ друзів. Проте, вони не можуть мати $k = 2$ друзів: якщо Лулу захирається в місті 1, а Танака в місті 5, тоді двох друзів недостатньо, щоб заблокувати дороги та щоб Лулу не могла дібратись до Танаки.

Кожне місто має різну загальну видимість (деякі міста знаходяться на більшій висоті та можуть бути побачені з більшої кількості місць). Позначимо v_1, v_2, \dots, v_n як видимості міст.

Бувши дуже відданою хованкам, Лулу вже спостерігала як проходить звичайний день в Info(1)cup Королівстві і знає, що під час дня видимість міст може змінюватись (відносно позиції сонця, трафіку і т.д.). Більш точно, Лулу хоче обробити q різних подій. Кожна подія впливає на одне місто. Якщо подія впливає на місто i , тоді v_i стає рівним

$$\left(v_i + \sum_{j \in \text{neighbours}(i)} v_j \right) \bmod 998244353.$$

Для кожної події, вона зацікавлена в значенні v_i **після операції**.

Interaction Protocol

Ваше рішення має включати "hideandseek.h" хедер.

```
#include "hideandseek.h"
```

Також ваше рішення має реалізовувати наступні функції:

```
void init(int n, int m, int k, int q, int v[], int X[], int Y[]);  
int update(int i);
```

Функції `init` задається n (кількість міст), m (кількість доріг), k (кількість друзів Лулу та Танаки), q (кількість подій) та масив v , де $v[i]$ – початкова видимість міста i для $1 \leq i \leq n$, а також два масиви X та Y , які позначають що є ребро між $X[i]$ та $Y[i]$, де $1 \leq i \leq m$. Ця функція викликається рішенням журі рівно один раз, перед усіма викликами `update`.

Функції `update` задається i , що позначає номер міста, на яке впливає подія (дивіться формулу вище). Функція має повертати значення $v[i]$ після оновлення. Учасник не має реалізовувати функцію `main`. Це буде реалізовано в файлі журі `grader.cpp`; вам буде дано приклад `grader.cpp` у вкладених файлах. Вам також буде надано файл `hideandseek.cpp` з простою незакінченою реалізацією.

Наша функція `main` буде читати з першого рядку стандартного вводу цілі числа n , m та k . Другий рядок містить послідовність з n чисел початкових значень. Наступні m рядків містять пари чисел u, v , що позначають, що є двостороння дорога між містами u та v . Рядок з номером $m + 3$ містить єдине число q . Наступний рядок містить q чисел, що позначають події.

Зверніть увагу, що учаснику дозволяється використовувати глобальні змінні, додаткові функції, методи та класи.

Restrictions

- $1 \leq n \leq 5 \cdot 10^5$
- $1 \leq m \leq 6 \cdot 10^5$
- $1 \leq q \leq 2,5 \cdot 10^6$
- $1 \leq k \leq 5$
- $0 \leq v_i < 998244353$

#	Points	Restrictions
1	23	$1 \leq n, q \leq 2000$
2	21	$1 \leq n, m, q \leq 2 \cdot 10^5$
3	25	$k = 1$
4	31	Без додаткових обмежень.

Examples

Input file	Output file	Explanations
5 6 3 1 2 3 4 5 1 2 1 3 1 4 2 5 3 5 4 5 3 1 5 3	10 14 27	<p>The network of towns is the one represented in the diagram below. After the first update, the visibility of town 1 becomes $1 + 2 + 3 + 4 = 10$. After the second update, the visibility of town 5 becomes $5 + 2 + 3 + 4 = 14$. After the third update, the visibility of town 3 becomes $3 + 10 + 14 = 27$.</p> 