

Задача C12. HAPPY

⌚ 3 сек. 📄 512 MB

Автор: Timur Degteari

“Happy” е популярна верига ресторанти в България. Мария е пътувала от Бургас до съседния град Варна – родния град на “Happy”. Тя среща приятелите си и отиде до един от ресторантите да отпразнува скорошния си успех на IATI. Известна с отличното си обслужване, сервитюрката даде на Мария предизвикателство – ако го разреши, поръчката ѝ ще бъде безплатна.

Има N Happy ресторанта във Варна, номерирани от 0 до $N - 1$. Те са свързани от $N - 1$ двупосочни пътища, като всеки път има определено количество щастлива енергия. Дадени са два ресторанта x и y , дефинираме *прост път* между тях като последователност от пътища, започващи от x и завършващи в y без повтарящи се ресторанти. Ресторантите са свързани, така че да има точно един *прост път* между всяка двойка ресторанти.

Поради някаква незнана причина, *щастие*то на *прост път* е дефинирано като **побитов XOR** на щастливите енергии на пътищата в него.

Задачата е Мария да намери сумата от *щастия*та на *простите пътища* между всяка двойка от отворени “Happy” ресторанти. Първоначално всички ресторанти са затворени, но имаме Q актуализации от два вида:

- вид 1 – дадени са цели числа $l[j]$ и $r[j]$, такива че $0 \leq l[j] \leq r[j] \leq N - 1$, всички ресторанти с номера от интервала $[l[j], r[j]]$ стават отворени (ако вече са били отворени, те си остават отворени);
- вид 2 – дадени са цели числа $l[j]$ and $r[j]$ такива че $0 \leq l[j] \leq r[j] \leq N - 1$, всички ресторанти с номера в интервала $[l[j], r[j]]$ стават затворени (ако някои от тях вече са били затворени, те си остават затворени);

Помогнете на Мария да реши тази задача!

Детайли по имплементацията

Трябва да имплементирате функция `happy`:

```
std::vector<long long int> happy (int N, int Q,  
    std::vector<int> u, std::vector<int> v, std::vector<int> h,  
    std::vector<int> t, std::vector<int> l, std::vector<int> r)
```

- N : броят на ресторантите Happy;
- Q : броят на актуализациите;
- u , v и h : вектори с по $N - 1$ числа. Всяка съответна тройка числа от векторите представя два ресторанта и щастливата енергия на пътя между тях;
- t , l , r : са вектори с по Q числа. Всяка тройка съответни елементи представлява вид на актуализацията, ляв и десен край на интервала ѝ.

Тази функция ще бъде извиквана веднъж за всеки тест и трябва да връща вектор с Q числа – сумата от *щастие*то на *простите пътища* между всяка двойка отворени ресторанти след всяка актуализация.

**XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
 ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026**

Ограничения

- $1 \leq N \leq 300\,000$;
- $1 \leq Q \leq 300\,000$;
- $1 \leq h[i] < 2^{20}$ за всяко $0 \leq i < N - 1$;
- Има точно един *прост път* между всяка двойка ресторанти.

Подзадачи

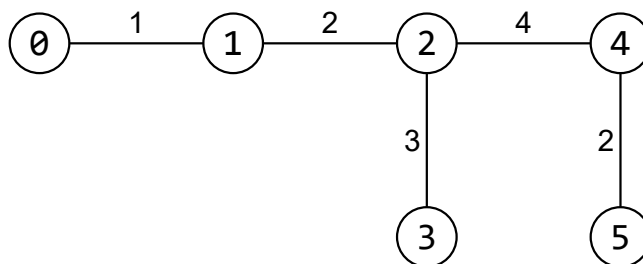
Подзадача	Точки	Необходимите подзадачи	N	Q	Други ограничения
0	0	—	—	—	Примерният тест.
1	7	0	≤ 100	≤ 100	-
2	8	0 – 1	$\leq 2\,000$	$\leq 2\,000$	-
3	19	—	$\leq 300\,000$	$\leq 300\,000$	$l[j] = r[j]$ за всички $0 \leq j < Q$ и $u[i] = i$, $v[i] = i + 1$ за всички $0 \leq i < N - 1$.
4	19	—			$l[j] = r[j]$ за всички $0 \leq j < Q$ и $h[i] = 1$ за всички $0 \leq i < N - 1$.
5	10	0	$\leq 300\,000$	≤ 10	-
6	17	—	$\leq 300\,000$	$\leq 300\,000$	Ако $t[j] = 1$, тогава ресторантите в интервала $[l[j], r[j]]$ са били затворени преди актуализацията; ако $t[j] = 2$, то ресторантите $[l[j], r[j]]$ са били отворени преди актуализацията ($0 \leq j < Q$).
7	20	0 – 6			-

Пример

Примерно извикване на функцията при $N = 6$ и $Q = 5$

happy(6, 5, {0, 1, 2, 2, 4}, {1, 2, 3, 4, 5}, {1, 2, 3, 4, 2},
 {1, 1, 2, 2, 1}, {1, 2, 2, 4, 0}, {2, 3, 2, 5, 5})

XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026



5-те актуализации са следните:

- 1 1 2 (отварят ресторантите 1 и 2) - след актуализацията имаме само *прост път* $1 \stackrel{2}{=} 2$ с щастие 2. Сумата е 2;
- 1 2 3 (отварят ресторантите 2 и 3) - след актуализацията имаме *простите пътища* между ресторантите 1 и 2 ($1 \stackrel{2}{=} 2$), 1 и 3 ($1 \stackrel{2}{=} 2 \stackrel{3}{=} 3$), и 2 и 3 ($2 \stackrel{3}{=} 3$) с щастията 2, $2 \wedge 3 = 1$ и 3 (\wedge е C++ оператор за побитово XOR). Сумата е $2 + 1 + 3 = 6$;
- 2 2 2 (затваря ресторант 2) - след актуализацията имаме *прост път* между ресторантите 1 и 3. Сумата е 1;
- 2 4 5 (затварят ресторанти 4 и 5) - след актуализацията имаме *прост път* между ресторантите 1 и 3. Сумата е 1;
- 1 0 5 (ресторантите от 0 до 5 отварят) - след актуализацията имаме *спрости пътища* всички двойки ресторанти. Сумата е 56.

Следователно, извикването трябва да върне $\{2, 6, 1, 1, 56\}$.

Формат на грейдъра

Формат на входа:

- ред 1: две цели числа - стойностите на N и Q .
- ред от $2 + i$ до $2 + N - 2$: три цели числа $u[i]$, $v[i]$, $h[i]$ - път между ресторантите с номера $u[i]$ и $v[i]$ и щастлива енергия $h[i]$.
- ред от $N + 1 + j$ до $N + 1 + Q - 1$: три цели числа $t[j]$, $l[j]$, $r[j]$ - актуализация от вид $t[j]$ за интервал от ресторанти с номера от $l[j]$ до $r[j]$.

Формат на изхода:

- ред i : i -тата стойност върната от извикването.