



ВИРТУАЛНО ДЪРВО

НАЦИОНАЛНА ЛАГЕР-ШКОЛА ПО ИНФОРМАТИКА,
19 АВГУСТ 2025 Г., ВРАЦА

Изготвил: Илиян Йорданов Йорданов

СТАНДАРТНА ПОСТАНОВКА

Дадено е дърво с N върха.

Интересуваме се многократно от различни подмножества на върховете.



ПОДГОТВИТЕЛНА ЗАДАЧА – СК2. CHASE, КОНТРОЛНО №3, 2023 ГОДИНА

Дадено е дърво с N върха и корен връх 1, като всеки връх има стойност A_i . Търсим най-дългия прост път без “завой”, за който НОД от стойностите е повече от 1.

Ограничения: $N \leq 10^5$, $A_i \leq 2 \cdot 10^6$.

[Линк](#) към условието.



ПОДГОТВИТЕЛНА ЗАДАЧА – СК2. CHASE, КОНТРОЛНО №3, 2023 ГОДИНА

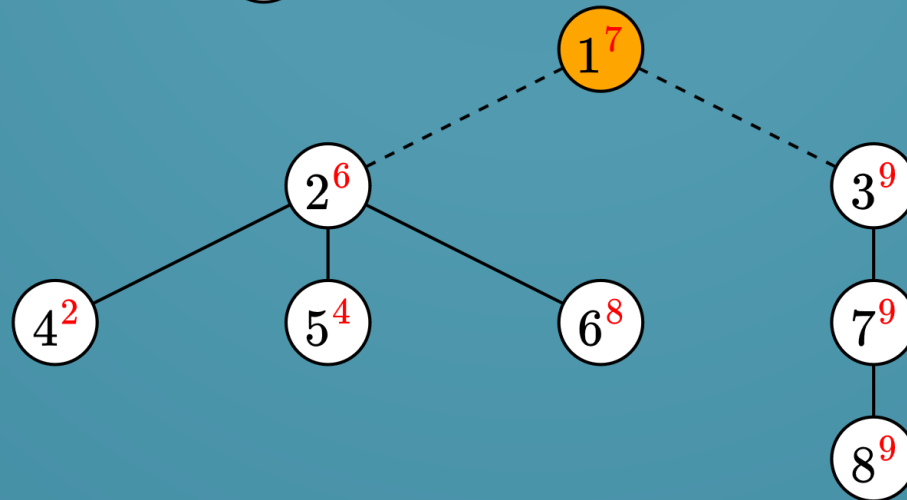
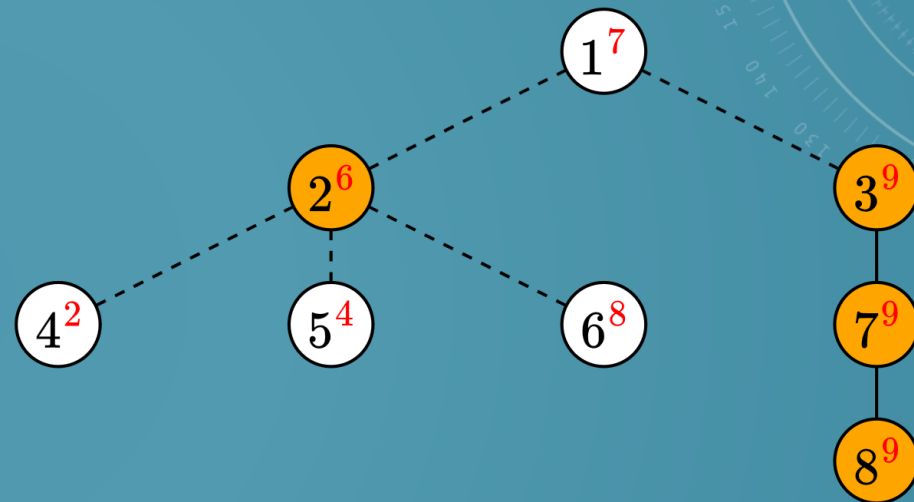
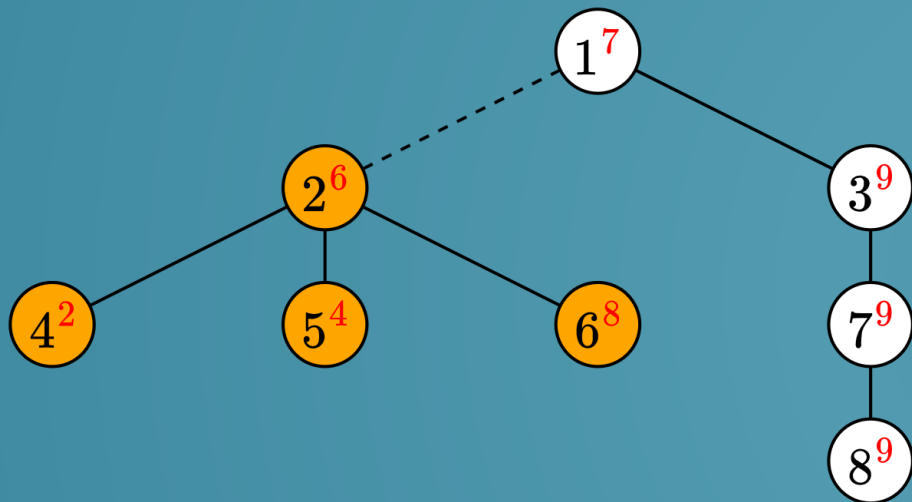
Дадено е дърво с N върха и корен връх 1, като всеки връх има стойност A_i . Търсим най-дългия прост път без “завой”, за който НОД от стойностите е повече от 1.

Ограничения: $N \leq 10^5$, $A_i \leq 2 \cdot 10^6$.

Решение: Използваме определението за НОД и от него следва, че е достатъчно да разгледаме по отделно върховете с число, делящо се на фиксирана стойност или по-добре на фиксирано просто число. Намираме дължината на най-дългите пътища с динамично в получените компоненти.



ПОДГОТВИТЕЛНА ЗАДАЧА – СК2. CHASE, КОНТРОЛНО №3, 2023 ГОДИНА



ПОДГОТВИТЕЛНА ЗАДАЧА – СК2. CHASE, КОНТРОЛНО №3, 2023 ГОДИНА

Дадено е дърво с N върха, като всеки връх има стойност A_i . Търсим най-дългия прост път без “завой”, за който НОД от стойностите е повече от 1.

Ограничения: $N \leq 10^5$, $A_i \leq 2 \cdot 10^6$.

Решение: Използваме определението за НОД и от него следва, че е достатъчно да разгледаме по отделно върховете с число, делящо се на фиксирана стойност или по-добре на фиксирано просто число. Намираме дължината на най-дългите пътища с динамично в получените компоненти.

Какви са сложностите при двата подхода?



ПОДГОТВИТЕЛНА ЗАДАЧА – СК2. CHASE, КОНТРОЛНО №3, 2023 ГОДИНА

Дадено е дърво с N върха, като всеки връх има стойност A_i . Търсим най-дългия прост път без “завой”, за който НОД от стойностите е повече от 1.

Ограничения: $N \leq 10^5$, $A_i \leq 2 \cdot 10^6$.

Решение: Използваме определението за НОД и от него следва, че е достатъчно да разгледаме по отделно върховете с число, делящо се на фиксирана стойност или по-добре на фиксирано просто число. Намираме дължината на най-дългите пътища с динамично в получените компоненти.

Сложността за първия подход е $O(N^3 \sqrt{MAX})$, а за втория подход е $O(N \log MAX)$.



ИЗВОДИ ОТ ЗАДАЧАТА

Подмножествата върхове разглеждахме само като свързани компоненти.

Затова нямаше нужда от техниката виртуално дърво.

Виртуално дърво се налага, ако разглеждаме подмножествата в цялост.



ПРИМЕРНА ЗАДАЧА – AT22. CHASE, ТРЕНИРОВЪЧНО №2, БАНКЯ, 2024 ГОДИНА

Дадено е дърво с N върха, като всеки връх има уникална стойност A_i . Търсим най-дългия прост път, за който НОД от стойностите на началото и края е повече от 1.

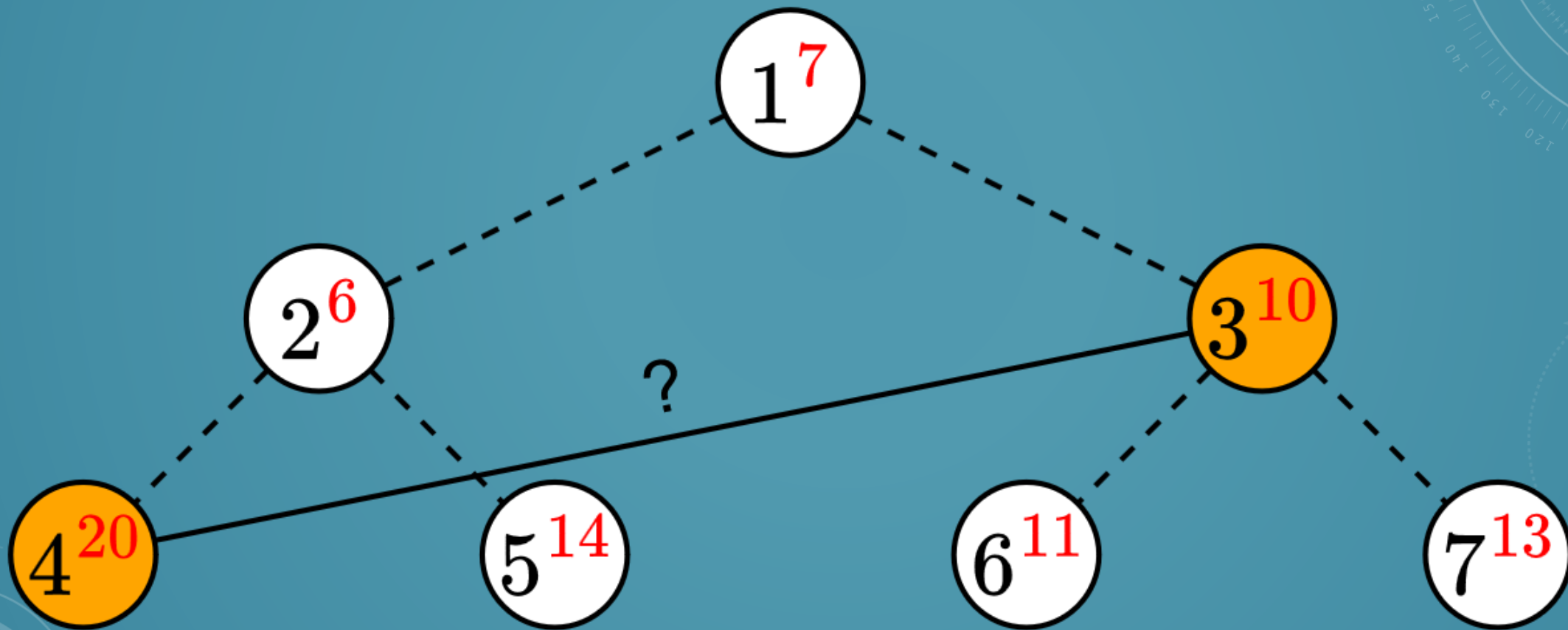
Ограничения: $N \leq 4 \cdot 10^5$, $A_i \leq 4 \cdot 10^5$.

[Линк](#) към условието.

Отново ще коренуваме дървото във връх 1.



ПРИМЕРНА ЗАДАЧА – AT22. CHASE, ТРЕНИРОВЪЧНО №2, БАНКЯ, 2024 ГОДИНА

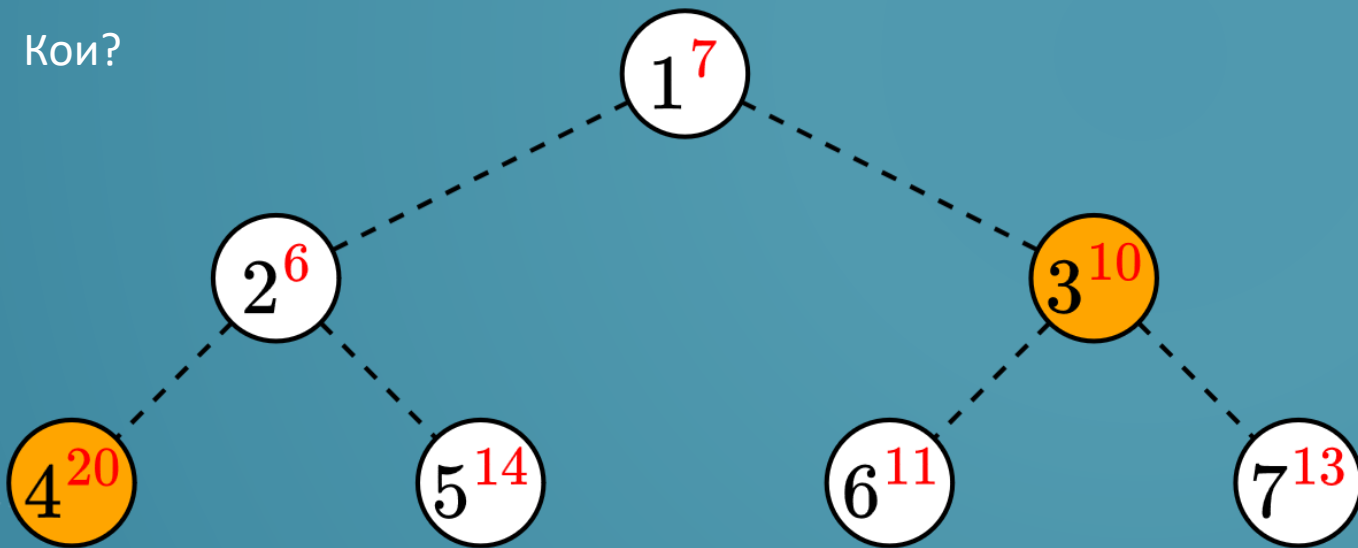


ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

Можем да го разглеждаме като дърво, наподобяващо по топология началното кореново дърво, и включващо всички върхове от подмножеството.

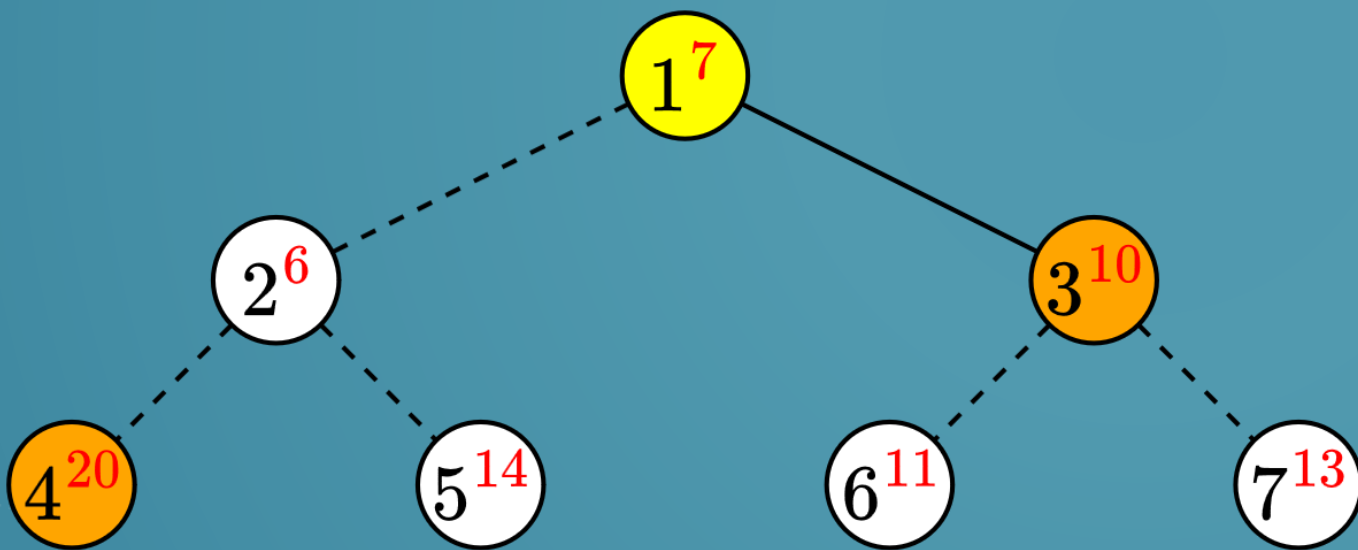
Понякога се налага да добавим още върхове от началното дърво.

Кои?



ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

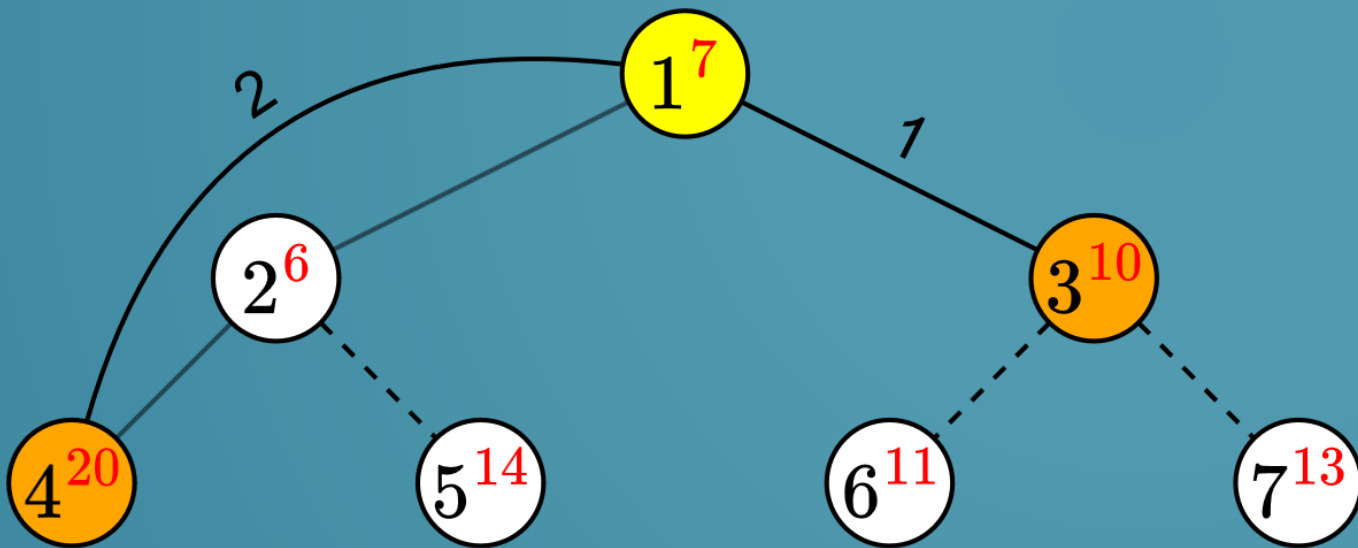
Ако върховете от подмножеството са v_1, v_2, \dots, v_k , то е достатъчно да добавим всички $LCA(v_i, v_j)$.



ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

Ако върховете от подмножеството са v_1, v_2, \dots, v_k , то е достатъчно да добавим всички $LCA(v_i, v_j)$.

Трябва да свържем всички върхове, което налага компресиране на някои пътища в ребра според задачата.



ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

Ако върховете от подмножеството са v_1, v_2, \dots, v_k , то $|\{LCA(v_i, v_j) | 1 \leq i, j \leq k\}|$ е $O(k)$.

Защо?



ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

Ако върховете от подмножеството са v_1, v_2, \dots, v_k , то $|\{LCA(v_i, v_j) | 1 \leq i, j \leq k\}| \leq k - 1$.

БОО върховете са наредени по *in* време. Тогава може да се докаже, че $\{LCA(v_i, v_j) | 1 \leq i, j \leq k\} = \{LCA(v_i, v_{i+1}) | 1 \leq i \leq k - 1\}$.



ПОСТРОЯВАНЕ НА ВИРТУАЛНОТО ДЪРВО

Ако върховете от подмножеството са v_1, v_2, \dots, v_k , то $|\{LCA(v_i, v_j) | 1 \leq i, j \leq k\}| \leq k - 1$.

БОО върховете са наредени по *in* време. Тогава може да се докаже, че $\{LCA(v_i, v_j) | 1 \leq i, j \leq k\} = \{LCA(v_i, v_{i+1}) | 1 \leq i \leq k - 1\}$.

Използвайки намирането на LCA чрез Ойлеровото обхождане на дървото, то $LCA(v_i, v_j)$ е някое измежду $LCA(v_i, v_{i+1}), LCA(v_{i+1}, v_{i+2}), \dots, LCA(v_{j-1}, v_j)$.



АЛГОРИТЪМ ЗА ВИРТУАЛНО ДЪРВО

- 1) Сортираме върховете от заявката по in време във възходящ ред. Нека след това върховете са v_1, v_2, \dots, v_k .
- 2) Добавяме върховете $LCA(v_i, v_{i+1})$.
- 3) Сортираме отново всички върхове по in време във възходящ ред и премахваме повтарящите се.
- 4) Обхождаме получените върхове и свързваме всеки връх v с баща си u във виртуалното дърво и компресиране пътя от v до u в едно ребро.

Как намираме бащите u ?



АЛГОРИТЪМ ЗА ВИРТУАЛНО ДЪРВО

- 1) Сортираме върховете от заявката по in време във възходящ ред. Нека след това върховете са v_1, v_2, \dots, v_k .
- 2) Добавяме върховете $LCA(v_i, v_{i+1})$.
- 3) Сортираме отново всички върхове по in време във възходящ ред и премахваме повтарящите се. Така имаме in обхождането на върховете.
- 4) Обхождаме получените върхове и поддържаеме в стек предшествениците. За всеки връх v (без първия, който е корен на виртуалното дърво) премахваме всички върхове в стека, които не са му предшественици. Тогава свързваме v с баща си u във виртуалното дърво, който е във върха на стека, и компресиране пътя от v до u в едно ребро. Добавяме v към стека.



АЛГОРИТЪМ ЗА ВИРТУАЛНО ДЪРВО

За 3) е удобно да използваме следното:

```
vers.erase(unique(vers.begin(), vers.end()), vers.end());
```

Заради 4) ни трябва *LCA* с двоично повдигане, за да правим и компресията на пътищата.

Сложност: $O(k \log N)$, а ако трябва само да свържем върховете е $O(k \log k)$.



ПРИМЕРНА ЗАДАЧА – AT22. CHASE, ТРЕНИРОВЪЧНО №2, БАНКЯ, 2024 ГОДИНА

Дадено е дърво с N върха, като всеки връх има уникална стойност A_i . Търсим най-дългия прост път, за който НОД от стойностите на началото и края е повече от 1.

Ограничения: $N \leq 4 \cdot 10^5$, $A_i \leq 4 \cdot 10^5$.

Решение: Достатъчно е за всяко просто число да построим виртуално дърво за върховете със стойност, делима се на него. При компресията заменяме всеки път с броя оригинални ребра в него. Накрая намираме големината на диаметъра и отговорът е дължината на най-големия намерен диаметър.

Сложност: $O(N \log N \log \log MAX)$.



ЗАДАЧА АЗ. BREAKDOWN, ЗИМЕН ТУРНИР, 2011 ГОДИНА

Дадено е претеглено дърво с N върха. Имаме Q заявки, като за всяка търсим най-малката сума ребра, които да премахнем, за да отделим даден връх от някои други.

Ограничения: $N \leq 2,5 \cdot 10^5$, сумата от броя върхове в заявките $\leq 5 \cdot 10^5$.

[Линк](#) към условието.



ЗАДАЧА А3. BREAKDOWN, ЗИМЕН ТУРНИР, 2011 ГОДИНА

Дадено е претеглено дърво с N върха. Имаме Q заявки, като за всяка търсим най-малката сума ребра, които да премахнем, за да отделим даден връх от някои други.

Ограничения: $N \leq 2,5 \cdot 10^5$, сумата от броя върхове в заявките $\leq 5 \cdot 10^5$.

Решение: За всяка заявка построяваме виртуалното дърво като за компресия използваме дължината на съответния път. Коренуваме всяко виртуално дърво във върха, който искаме да отделим. След това с динамично в дървото намираме отговора, като внимаваме за двата вида върхове – тези от заявката и допълнителните LCA върхове.

Сложност: $O(MAXS \log MAXS)$.



ЗАДАЧА 1292D. CHAOTIC V., CODEFORCES

Разглеждаме дърво с върхове естествените числа, като корен е числото 1. Бащата на всеки връх v се получава, като го разделим на най-малкия прост делител. Дадени са N върха – $k_1!, k_2!, \dots, k_N!$ и разглеждаме връх на дървото, който минимизира сумата от разстоянията до дадените върхове. Търсим тази минимална сума.

Ограничения: $N \leq 10^6$, $k_i \leq 10^3$ (може и до 10^5).

[Линк](#) към условието.



ЗАДАЧА 1292D. CHAOTIC V., CODEFORCES

Разглеждаме дърво с върхове естествените числа, като корен е числото 1. Бащата на всеки връх v се получава, като го разделим на най-малкия прост делител. Дадени са N върха – $k_1!, k_2!, \dots, k_N!$ и разглеждаме връх на дървото, който минимизира сумата от разстоянията до дадените върхове. Търсим тази минимална сума.

Ограничения: $N \leq 10^6$, $k_i \leq 10^3$ (може и до 10^5).

Решение: Нека се пробваме да построим виртуалното дърво от дадените върхове. Ако имаме виртуалното дърво, то можем да намерим негов центроид (относно дадените върхове), като някои върхове могат да имат тегло повече от 1 (ако се срещат няколко пъти в N -те).

Как построяваме виртуалното дърво?



ЗАДАЧА 1292D. CHAOTIC V., CODEFORCES

За да сравняваме два върха по in време е достатъчно да сравняваме лексикографски степените на простите им делители в каноничното разлагане – от най-големите до най-малките. Това означава, че факториелите се подреждат по големина спрямо основата.

Можем да забележим, че $LCA(x!, y!)$ за $x < y$ е число, което се получава като “суфикс” на каноничното разлагане на $x!$, започващ от най-големия прост делител на $(x + 1)(x + 2) \dots y$.

Но как сравняваме лесно по in време всички тези върхове накрая?



АЛТЕРНАТИВЕН АЛГОРИТЪМ ЗА ВИРТУАЛНО ДЪРВО

- 1) Сортираме върховете от заявката по in време във възходящ ред. Нека след това върховете са v_1, v_2, \dots, v_k .
- 2) Комбиниране добавянето на $LCA(v_i, v_{i+1})$ и добавянето на ребрата, като се възползваме от подредения ред на v_1, v_2, \dots, v_k . Отново поддържаме стек с предшествениците.
- 3) Когато сме на връх v_i , първо трябва да добавим $LCA(v_{i-1}, v_i)$. Този връх се намира преди v_i , както и преди v_{i-1} , така че просто трябва да намерим мястото му в стека. (възможно е да се е срещал преди)
- 4) Когато махаме връх в стека, трябва да се грижим да го свържем с баща му във виртуалното дърво. Трябва да компресиране и пътя между двата върха.
- 5) Добавяме връх v_i в стека.
- 6) Накрая изпразваме стека и навързваме останалите върхове там.



ЗАДАЧА 1292D. CHAOTIC V., CODEFORCES

Разглеждаме дърво с върхове естествените числа, като корен е числото 1. Бащата на всеки връх v се получава, като го разделим на най-малкия прост делител. Дадени са N върха – $k_1!, k_2!, \dots, k_N!$ и разглеждаме връх на дървото, който минимизира сумата от разстоянията до дадените върхове. Търсим тази минимална сума.

Ограничения: $N \leq 10^6$, $k_i \leq 10^3$ (може и до 10^5).

Решение: Нека се пробваме да построим виртуалното дърво от дадените върхове. Ако имаме виртуалното дърво, то можем да намерим негов центроид (относно дадените върхове), като някои върхове могат да имат тегло повече от 1 (ако се срещат няколко пъти в N -те).

Използвайки алтернативния алгоритъм за виртуално дърво, характеризираме всеки връх само с дълбочината (което е просто сумата от степените в каноничното разлагане), за да преценяваме дали съответния LCA връх е предшественик на даден връх.

Сложност: $O(MAXK \log MAXK \log \log MAXK + N)$.



ДОМАШНО (ЗАДАЧИ ЗА УПРАЖНЕНИЕ)

- Дорешаване на задача A3. breakdown, Зимен турнир, 2011 година - [линк](#) към условието.
- Дорешаване на задача 1292D. Chaotic V., Codeforces - [линк](#) към условието.
- Задача 613D. Kingdom and its Cities, Codeforces - [линк](#) към условието.
- Задача Adjacent Leaves, CodeChef - [линк](#) към условието.





БЛАГОДАРЯ ЗА
ВНИМАНИЕТО!

Изготвил: Илиян Йорданов Йорданов