Jacek Tomasiewicz
Treść zadania, Opracowanie

Bartłomiej Dudek
Program

Dostępna pamięć: 128 MB.

OI, etap III, dzień pierwszy, 2.04.2014

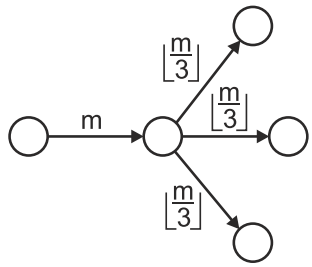
Mrowisko

Mrówki plądrują opuszczone mrowisko w poszukiwaniu jedzenia. Mrowisko składa się z n komór oraz łączących je $n-1$ korytarzy. Wiemy, że z każdej komory do każdej innej komory można przejść w dokładnie jeden sposób. Inaczej mówiąc, komory i korytarze tworzą drzewo.

We wszystkich komorach, do których prowadzi tylko jeden korytarz, znajdują się wejścia do mrowiska. Przy każdym z wejść ustawiło się po g grup mrówek składających się kolejno z $m_1, m_2, \ldots, m_g$ mrówek. Grupy będą wchodziły do mrowiska pojedynczo (kolejna grupa wchodzi dopiero wtedy, gdy w mrowisku nie ma już żadnych mrówek). Wewnątrz mrowiska mrówki poruszają się w określony sposób:

- *Po wejściu do komory, w której zbiega się d nieodwiedzonych jeszcze przez daną grupę mrówek korytarzy, grupa ta dzieli się na d równolicznych grup. Każda powstała w ten sposób grupa porusza się dalej jednym z tych d korytarzy. Jeśli $d = 0$, to grupa mrówek po prostu opuszcza mrowisko.*
- *Jeśli mrówki nie mogą podzielić się na równoliczne grupy, to silniejsze osobniki zjadają słabsze, do momentu, aż będzie możliwy podział na grupy równej wielkości (w szczególności, jedna mrówka może zjeść samą siebie i zniknąć).*

Poniższy rysunek przedstawia grupę m mrówek, które wchodzą do komory, w której zbiegają się trzy nieodwiedzone jeszcze korytarze, a następnie dzielą się na trzy równe grupy o licznościach $\lfloor m/3 \rfloor$.



Nad jednym z korytarzy znajduje się otwór, przez który do środka dostał się długi język głodnego mrówkojada. Wiemy, że mrówkojad zjada każdą przechodzącą tym korytarzem grupę mrówek, która składa się z dokładnie k mrówek. Chcemy wiedzieć, ile mrówek zje mrówkojad.

Wejście

Pierwszy wiersz standardowego wejścia zawiera trzy liczby całkowite n, g, k ($2 \leqslant n, g \leqslant 1\,000\,000$, $1 \leqslant k \leqslant 10^9$) pooddzielane pojedynczymi odstępami. Oznaczają one odpowiednio liczbę komór, liczbę grup mrówek oraz liczność grup mrówek zjadanych przez mrówkojada. Komory są ponumerowane liczbami od 1 do n.

Drugi wiersz zawiera g liczb całkowitych $m_1, m_2, \ldots, m_g$ ($1 \leqslant m_i \leqslant 10^9$) pooddzielanych pojedynczymi odstępami, gdzie m_i oznacza liczność i-tej grupy mrówek czekającej przy każdym z wejść do mrowiska. Kolejnych $n-1$ wierszy opisuje korytarze mrowiska; i-ty z nich zawiera dwie liczby całkowite a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) oddzielone pojedynczym odstępem, oznaczające, że komory o numerach a_i i b_i są połączone korytarzem. Język mrówkojada znajduje się w korytarzu, którego opis pojawia się jako pierwszy na wejściu.

W testach wartych 50% punktów liczba wszystkich grup mrówek wchodzących do mrowiska nie przekroczy 1 000 000. Ponadto w podzbiorze tych testów wartym 20% punktów zachodzi dodatkowy warunek $n, g \leqslant 100$.

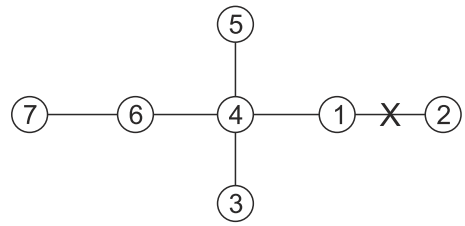
Wyjście

Twój program powinien wypisać na standardowe wyjście jeden wiersz zawierający jedną liczbę całkowitą, oznaczającą liczbę mrówek, które zje mrówkojad.

Przykład

Dla danych wejściowych:

```
7 5 3
3 4 1 9 11
1 2
1 4
4 3
4 5
4 6
6 7
```



poprawnym wynikiem jest:

```
21
```

Wyjaśnienie do przykładu: *Przy każdej z komór o numerach 2, 3, 5 i 7 ustawia się po 5 grup mrówek. Mrówkojad zje 3 mrówki z pierwszej grupy startującej z komory 2 oraz po 3 mrówki z czwartych i piątych grup startujących z komór 3, 5 i 7.*

Testy „ocen":

`1ocen`: *$n = 20$, $g = 20$, $k = 5$, komory w mrowisku są połączone korytarzami w jeden długi tunel. Język mrówkojada znajduje się w krańcowym korytarzu. Rozmiary grup to $1, \ldots, 20$.*

`2ocen`: *$n = 2^{19} + 1$, $g = 20$, $k = 1$, budowa mrowiska jest następująca: komora 1 jest połączona z komorą n (w tym korytarzu znajduje się mrówkojad), a komora i-ta (dla $i = 2, 3, \ldots, n-1$) jest połączona z komorą $\left\lfloor \frac{i}{2} \right\rfloor$. Do mrowiska wchodzą grupy mrówek o rozmiarach będących kolejnymi potęgami dwójki: $2^0, 2^1, \ldots, 2^{19}$.*

Rozwiązanie

Mrowisko w zadaniu opisane jest jako graf będący drzewem, po którym chodzą grupy mrówek. Każda z grup startuje kolejno w każdym z liści i w każdym wierzchołku dzieli się na x grup, gdzie $x+1$ to stopień tego wierzchołka. Wyjątkiem są wierzchołki startowe, w których grupy nie dzielą się. Dzielenie jest całkowitoliczbowe i każda z mniejszych grup wyrusza jedną z krawędzi, z pominięciem tej, którą przyszła. Gdy jakaś grupa dojdzie do liścia, to znika.

W drzewie zaznaczamy jedną krawędź i musimy policzyć, ile grup o liczności k przez nią przeszło. Oznaczmy końce tej krawędzi przez a i b.

Zanim zaczniemy myśleć nad różnymi rozwiązaniami, zauważmy, że dla każdej grupy mrówek wchodzącej do mrowiska, co najwyżej jedna grupa mrówek pochodząca z tej początkowej grupy przejdzie obok mrówkojada. Jest tak dlatego, iż mrówki z danej grupy idą zawsze nieodwiedzoną krawędzią, do momentu aż ich grupa będzie za mała, żeby się podzielić, lub dojdą do liścia. Z tej obserwacji będą korzystały wszystkie przedstawione rozwiązania.

Liczba wierzchołków drzewa to n, a wszystkich grup mrówek jest g. Dodatkowo, przez m oznaczymy górne ograniczenie na liczność grupy, czyli dokładniej $m = \max(m_1, m_2, \ldots, m_g)$.

Rozwiązanie siłowe $O(n^2 \cdot g)$

Najprostszym rozwiązaniem jest symulacja ruchu każdej grupy mrówek z każdego liścia. Wszystkich grup jest g, liczba liści i rozmiar drzewa jest rzędu $O(n)$, więc złożoność czasowa wyniesie $O(n^2 \cdot g)$.

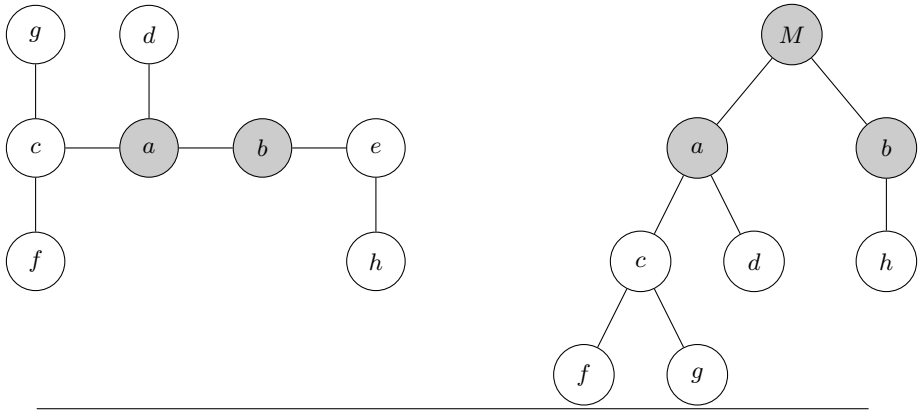
Za takie rozwiązanie można było uzyskać około 20% punktów. Implementacje znajdują się w plikach `mros1.cpp` oraz `mros4.pas`.

Rozwiązanie wolne $O(n \cdot g \cdot \log m)$

Spróbujmy skompresować wejściowe drzewo. Zauważmy, że możemy usunąć wierzchołki (różne od a i b), z których wychodzą tylko dwie krawędzie. Takie wierzchołki nie zmieniają wyniku, ponieważ gdy grupa przez nie przechodzi, jej liczność nie zmienia się (dzielimy przez 1).

Następnie dla każdego wierzchołka powinniśmy umieć stwierdzić, którą krawędzią iść, aby dojść do mrówkojada (będzie tylko jedna taka krawędź, a wszystkie inne nas nie interesują, ponieważ nie wpływają na wynik). Takie kierunki możemy wyznaczyć, przykładowo, ukorzeniając nasze drzewo w miejscu mrówkojada (rys. 1). W ten sposób, poruszając się w górę drzewa, będziemy iść zawsze w kierunku mrówkojada.

Przypomnijmy, że m oznacza górne ograniczenie na liczność grupy. Chodzenie z każdego liścia, zawsze w stronę mrówkojada (dopóki rozmiar grupy jest niezerowy), będzie miało złożoność $O(\log m)$, gdyż w każdym kroku zmniejszamy rozmiar grupy przynajmniej dwukrotnie. Ostatecznie, złożoność całego rozwiązania wyniesie $O(n \cdot g \cdot \log m)$.



Rys. 1: Drzewo po prawej stronie powstaje przez ukorzenienie drzewa po lewej stronie w pozycji mrówkojada. W drzewie po prawej stronie usunęliśmy wierzchołek e, ponieważ wychodziły z niego tylko dwie krawędzie.

Za takie rozwiązanie można było uzyskać około 50% punktów, a implementacje znajdują się w plikach `mros2.cpp` oraz `mros5.pas`.

Rozwiązanie wzorcowe $O((n+g)\log g)$

Zastanówmy się, jak duża grupa mrówek powinna znaleźć się w wierzchołku a, aby mrówki, które udadzą się w kierunku wierzchołka b, zostały zjedzone przez mrówkojada.

Wyznaczanie przedziałów

Niech p_u oznacza liczbę grup, na które podzielą się mrówki, będąc w wierzchołku u. Wiemy, że mrówkojad zjada grupy złożone z k mrówek, a więc jeśli w wierzchołku a będzie dokładnie $p_a \cdot k$ mrówek, to po podziale zostanie ich dokładnie k. Ponieważ dzielimy całkowitoliczbowo, to możemy dodać do tego dowolną liczbę mniejszą od p_a, czyli wartości $p_a \cdot k + 1$, $p_a \cdot k + 2$, …, $p_a \cdot k + p_a - 1$ również będą poprawne. Ogólniej, dostajemy przedział liczb całkowitych od wartości $x = p_a \cdot k$ (włącznie) do $y = p_a \cdot (k+1)$ (wyłącznie). Taki przedział oznaczmy jako $[x..y)$.

Znaleźliśmy przedział określający liczbę mrówek, które powinny być w wierzchołku a. Cofnijmy się o jeden wierzchołek wcześniej, czyli weźmy pod uwagę sąsiadów wierzchołka a. Następnie dla nich spróbujmy znaleźć przedziały, takie że (po podzieleniu) do wierzchołka a przybędzie szukana liczba mrówek, czyli wartość z przedziału $[x..y)$ (co w efekcie spowoduje, że do mrówkojada dojdzie dokładnie k mrówek).

Dla ustalenia uwagi, rozpatrzmy jednego z sąsiadów, wierzchołek c. W tym wierzchołku mrówki podzielą się na p_c grup. W związku z tym, każda wartość z przedziału $[p_c \cdot x..p_c \cdot y)$ będzie poprawna. Zauważmy, że ponownie jest to spójny przedział.

W ten sposób uzyskujemy schemat konstrukcji przedziałów, który możemy powtarzać, dla coraz to bardziej oddalonych wierzchołków, aż dojdziemy do liści. Koszt

wyznaczenia przedziału dla każdego wierzchołka jest stały. Jeśli będziemy się poruszać od położenia mrówkojada do wszystkich innych wierzchołków (na przykład przeszukiwaniem w głąb), to każdy z nich odwiedzimy raz, więc w czasie $O(n)$ uzyskamy przedziały wartości dla wszystkich liści. Warto zauważyć, że jeśli w jakimś wierzchołku początek wyznaczonego przedziału jest większy niż m, to nie musimy już obliczać wyników dla wierzchołków w jego poddrzewie, gdyż w takim przypadku i tak żadna grupa mrówek z tego poddrzewa nie dotrze do mrówkojada.

Sprawdzanie grup mrówek

Znając te przedziały, umiemy odpowiadać w czasie stałym, czy grupa o rozmiarze w, startując z ustalonego liścia, zostanie zjedzona przez mrówkojada. Wystarczy sprawdzić, czy wartość w zawiera się w przedziale obliczonym dla tego liścia.

Jeśli wykonamy to dla wszystkich grup i wszystkich liści, to złożoność całego rozwiązania wyniesie $O(n \cdot g)$. Takie rozwiązanie otrzymywało 50% punktów i zostało zaimplementowane w plikach `mros3.cpp` oraz `mros6.pas`.

Przyspieszenie rozwiązania

Powyższe rozwiązanie można jeszcze usprawnić. Wystarczy posortować wszystkie rozmiary grup mrówek, a następnie dla każdego liścia wyszukać binarnie spójny podciąg grup mrówek, które zostaną zjedzone przez mrówkojada.

Sortowanie zajmuje czas $O(g \log g)$, a wyszukiwanie binarne $O(n \log g)$. Całkowita złożoność czasowa wyniesie zatem $O((n+g) \log g)$. Takie rozwiązanie zostało zaimplementowane w plikach `mro.cpp` oraz `mro1.pas`. Przy implementacji należy pamiętać, że wynik powinniśmy przechowywać w 64-bitowym typie zmiennych całkowitych. Ponadto polecamy Czytelnikowi zastanowienie się nad tym, dlaczego taki typ danych jest wystarczający (na pierwszy rzut oka nie jest to wcale oczywiste).

Testy

Każda grupa składała się z 5 testów. We wszystkich testach drzewo po jednej stronie mrówkojada zostało wygenerowane losowo, a po drugiej stronie znajdowało się:

- `a` – pełne drzewo binarne o korzeniu przy mrówkojadzie i z kilkoma małymi drzewami doczepionymi wewnątrz,
- `b` – pełne drzewo binarne, doczepione liściem do krawędzi z mrówkojadem,
- `c` – ścieżka długości $\frac{n}{3}$ z doczepionymi gwiazdami o rozmiarach około $\frac{n}{20}$,
- `d` – ścieżka długości $\frac{n}{6}$ z doczepionymi poddrzewami,
- `e` – $n-2$ wierzchołki doczepione bezpośrednio do krawędzi z mrówkojadem.