

5A – Zbiory niezależne

autor zadania: Marek Sokołowski

Treść

Mamy dane liczby ℓ , r oraz c . Naszym zadaniem jest policzyć ile istnieje różnych nieetykietowanych drzew, których wierzchołki są pomalowane na jeden z c kolorów i w których rozmiar maksymalnego zbioru niezależnego zawiera się w przedziale $[\ell, r]$. Wynik należy podać modulo 998 244 353.

Rozwiązanie

Wspomniana liczba pierwsza dla bardziej doświadczonych zawodników jest sygnałem, że będziemy intensywnie korzystać z szybkiej transformaty Fouriera, służącej do szybkiego mnożenia wielomianów. Mówiąc ściślej, będziemy korzystać z operacji na funkcjach tworzących: będziemy chcieli umieć je dodawać, mnożyć, ale także dla funkcji $F(x)$ będziemy chcieli umieć obliczać $\exp(F(x))$ i $\frac{1}{F(x)}$. Informacje na temat tego jak wykonywać te operacje w czasie $\mathcal{O}(n \cdot \log(n))$, gdzie n maksymalnym interesującym nas współczynnikiem, możliwe są do znalezienia w internecie.

Rozważmy najpierw zadanie, w którym chcemy liczyć drzewa o liczbie wierzchołków równej n , zamiast o rozmiarze zbioru niezależnego należącego do przedziału $[\ell, r]$, oraz interesują nas różne ukorzenione drzewa, a nie dowolne. Użyjmy zatem programowania dynamicznego: niech $L[i]$ będzie liczbą takich różnych nieetykietowanych ukorzenionych lasów, które zawierają dokładnie i wierzchołków. Niech $D[i]$ będzie liczbą ukorzenionych drzew o i wierzchołkach. $D[n]$ będzie zatem interesującym nas wynikiem. Programowanie dynamiczne inicjalizujemy oczywiście jako $L[0] = 1$. Zachodzi też własność $D[i] = c \cdot L[i-1]$ – drzewo możemy utożsamiać z lasem pozostałym po usunięciu jego korzenia, który to mógł mieć jeden z c kolorów. Warto zauważyć, że ciąg (L_n) możemy utożsamiać z jego funkcją tworzącą, którą będziemy oznaczać jako $L(x)$. Warto wprowadzić tutaj pomocnicze ciągi – niech $L_i(x)$ oznacza funkcję tworzącą dla ciągu, który oznacza liczbę lasów o odpowiedniej liczbie wierzchołków, w których każde drzewo ma rozmiar co najwyżej i . Zaczynamy oczywiście z $L_0(x) = 1$.

Jeśli poznamy wartość $D[i]$, to dość łatwo jest, znając również $L_{i-1}(x)$, obliczyć $L_i(x)$. Dla każdego drzewa liczonego w $D[i]$ możemy dowolną liczbę jego kopii dołożyć do dowolnego lasu. Dołożenie dowolnej liczby kopii drzewa o i wierzchołkach możemy wykonać poprzez przemnożenie przez $\sum_{j=0}^{\infty} x^{ij} = \frac{1}{1-x^i}$. Skoro dla każdego drzewa możemy niezależnie zdecydować o liczbie jego dołożonych kopii, to otrzymujemy równość $L_i(x) = L_{i-1}(x) \cdot \frac{1}{(1-x^i)^{D[i]}}$.

Łatwo się domyślić, że zamiast obliczać kolejne $L_i(x)$, będziemy utrzymywać jedną funkcję i ją aktualizować. Mnożenie jest jednak trudne i wolne, co możemy zatem zrobić zamiast niego? Chcąc finalnie obliczyć $\prod_{i=1}^r \frac{1}{(1-x^i)^{D[i]}}$, możemy zamiast tego obliczyć $\exp(\sum_{i=1}^r \log(\frac{1}{(1-x^i)^{D[i]}}))$. Dodawanie wydaje się już prostsze, jednak by upewnić się, że istnieje jakakolwiek nadzieja na poprawę czasową, należy spojrzeć na rozwinięcie Taylora dla funkcji $\log(\frac{1}{(1-x^i)^{D[i]}})$. Otrzymujemy:

$$\log\left(\frac{1}{(1-x^i)^{D[i]}}\right) = D[i] \cdot \sum_{j=1}^{\infty} \frac{x^{ij}}{j}$$

Jest to dla nas świetna wiadomość. W i -tym kroku, logarytm który dodajemy, jest nie tylko bardzo łatwy do obliczenia, ale również ma tylko $\lfloor \frac{r}{i} \rfloor$ niezerowych współczynników! Możemy zatem dodać wszystkie logarytmy w złożoności czasowej $\mathcal{O}(r \cdot \log(r))$, a następnie w tej samej złożoności przyłożyć do nich eksponentę. Jest jednak pewien problem. By obliczyć $D[i]$ musimy poznać $[x^{i-1}]L_{i-1}(x)$, a w tym celu, musielibyśmy dla każdego i po drodze obliczać eksponentę dość dużej funkcji tworzącej.

Potrzebujemy zatem sprytnego pomysłu. Powiedzmy, że obliczyliśmy już $\log(L_{2^k}(x))$ dla pewnego k . Możemy zatem pozwolić sobie na obliczenie $L_{2^k}(x)$. Spróbujmy obliczyć jednocześnie wszystkie wartości $D[i]$ dla i należącego do przedziału $[2^k + 1, 2^{k+1}]$. Jak możemy to zrobić? Oznaczmy:

$$\begin{aligned} P(x) &= L_{2^k}(x) \mod x^{2^k} = L(x) \mod x^{2^k} \\ S(x) &= \frac{L_{2^k}(x) - P(x)}{x^{2^k}} \mod x^{2^k} \\ W(x) &= \frac{L_{2^{k+1}}(x) - P(x)}{x^{2^k}} \mod x^{2^k} = \frac{L(x) - P(x)}{x^{2^k}} \mod x^{2^k} \end{aligned}$$

$P(x)$ i $W(x)$ odpowiadają zatem wszystkim lasom, których liczba wierzchołków mieści się w przedziałach odpowiednio $[0, 2^k)$ i $[2^k, 2^{k+1})$, zaś $S(x)$ odpowiada tym lasom, w których drzewa nie przekraczają rozmiarem 2^k . Korzystając z faktu, że w lesie o sumarycznej liczbie wierzchołków mniejszej niż 2^{k+1} zmieści się co najwyżej jedno drzewo o rozmiarze większym niż 2^k , możemy rozpisać równanie, dzięki któremu będziemy w stanie wyznaczyć $W(x)$:

$$\begin{aligned} W(x) &= S(x) + P(x) \cdot W(x) \cdot x \cdot c \mod x^{2^k} \\ W(x) &= \frac{S(x)}{1 - P(x) \cdot x \cdot c} \mod x^{2^k} \end{aligned}$$

Umiejąc wykonywać operacje na funkcjach tworzących możemy zatem obliczyć $W(x)$, dzięki czemu będziemy w stanie poznać $D[i]$ dla wszystkich i z przedziału $[2^k + 1, 2^{k+1}]$. Dla każdego z nich będziemy zatem mogli zaktualizować $L(x)$ o odpowiedni logarytm.

Ponieważ obliczenie kolejnych wartości wykonujemy w czasie $\mathcal{O}(2^k \cdot k)$, to obliczenie liczby drzew ukorzenionych o n wierzchołkach dla każdego n nie większego niż r możemy wykonać w czasie $\mathcal{O}(r \cdot \log(r))$. Pozostają dwa problemy: interesowały nas rozmiary zbiorów niezależnych, oraz drzewa nie miały być ukorzenione.

Z pierwszym problemem możemy poradzić sobie nieco komplikując obliczenia. Wiemy, że dla ukorzonego drzewa istnieje algorytm zachłanny, który oblicza rozmiar maksymalnego zbioru niezależnego – wierzchołek należy do zbioru niezależnego wtedy i tylko wtedy, gdy żadnego jego dziecka do niego nie należy.

Zamiast funkcji tworzącej $L(x)$, możemy zdefiniować funkcje tworzące $A(x)$ oraz $B(x)$. $[x^i]A(x)$ oznacza liczbę takich ukorzenionych lasów o sumarycznym zbiorze niezależnym rozmiaru i , w których żaden korzeń nie został wzięty do zbioru niezależnego przez algorytm zachłanny. $[x^i]B(x)$ oznacza liczbę wszystkich ukorzenionych lasów o sumarycznym zbiorze niezależnym rozmiaru i . Analogicznie do $L_i(x)$ możemy zdefiniować $A_i(x)$ i $B_i(x)$ – interesują nas jedynie ukorzenione lasy, w których żadne drzewo nie ma zbioru niezależnego o rozmiarze większym niż i . Okazuje się, że analogicznie do przypadku z ograniczeniem na liczbę wierzchołków, możemy znając $A_{2^k}(x)$ i $B_{2^k}(x)$ obliczyć $A_{2^{k+1}}(x)$ i $B_{2^{k+1}}(x)$ w czasie $\mathcal{O}(2^k \cdot k)$. Przekształcenia działają na tej samej zasadzie, jednak oczywiście są one bardziej skomplikowane, gdyż musimy rozwiązać układ dwóch równań z dwiema niewiadomymi. Zabawę z dokładnym przekształceniem wzorów, postanowiliśmy pozostawić jako ćwiczenie dla czytelnika.

Z drugim problemem, którym jest obliczenie liczby nieukorzenionych drzew, a nie ukorzenionych, pomoże nam pewien niesławny wzór.

Obserwacja (Otter's formula). *Ustalmy nieukorzenione nieetykietowane drzewo T . Niech v będzie liczbą różnych drzew ukorzenionych w wierzchołku i izomorficznych z T . Niech e będzie liczbą różnych drzew ukorzenionych w krawędzi i izomorficznych z T . Niech s będzie równe 1, jeśli T jest symetryczne względem którejkolwiek krawędzi, a 0 w przeciwnym przypadku. Wtedy $v - e + s = 1$.*

Korzystając z $A(x)$ oraz $B(x)$ obliczymy $G(x)$ oraz $H(x)$. Niech $[x^i]G(x)$ oznacza liczbę ukorzenionych drzew o rozmiarze zbioru niezależnego równym i , w których korzeń nie został wzięty przez algorytm zachłanny do zbioru niezależnego. Niech $[x^i]H(x)$ oznacza liczbę ukorzenionych drzew o rozmiarze zbioru niezależnego równym i , w których korzeń został wzięty przez algorytm zachłanny do zbioru niezależnego. Niech $[x^i]T(x)$ oznacza liczbę nieukorzenionych drzew o rozmiarze zbioru niezależnego równym i . Sumując powyższy wzór Ottera po wszystkich nieetykietowanych drzewach otrzymujemy:

$$T(x) = G(x) + H(x) - \frac{1}{2}G(x)^2 + \frac{1}{2}G(x^2) - G(x)H(x) - \frac{1}{2x}H(x)^2 + \frac{1}{2x}H(x^2)$$

$T(x)$ okazuje się być dokładnie funkcją tworzącą ciąg, o który jesteśmy pytani w zadaniu! Znając dla każdego n nie większego niż r liczbę drzew o rozmiarze zbioru niezależnego równym n , możemy po prostu zsumować te wartości dla wszystkich n z przedziału $[\ell, r]$. Otrzymujemy dzięki temu rozwiązanie działające w złożoności $\mathcal{O}(r \cdot \log(r))$ z ogromną stałą.