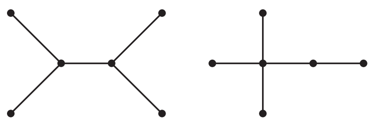
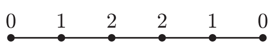
Informatyczny kącik olimpijski (22): Zliczanie planów sieci drogowej

Tytułowe zadanie pochodzi z finału Potyczek Algorytmicznych 2008. Należy w nim wyznaczyć liczbę różnych (nieetykietowanych) drzew n-wierzchołkowych o średnicy równej d i podać ją modulo pewna nieduża liczba pierwsza p. Dwa drzewa uznajemy przy tym za różne, jeżeli są nieizomorficzne, tzn. nie można wierzchołkom jednego z nich przyporządkować odpowiadających im wierzchołków drugiego tak, by były zachowane incydencje. Średnicą drzewa nazywamy długość (w sensie liczby krawędzi) najdłuższej ścieżki w drzewie. Na przykład dla $n = 6$, $d = 3$ istnieją dokładnie dwa różne drzewa:



Załóżmy na chwilę, że d jest nieparzyste. Wyobraźmy sobie jakieś drzewo o średnicy d z zaznaczoną ścieżką wyznaczającą tę średnicę. Teraz do każdego z wierzchołków na tej ścieżce można próbować doczepić jakieś gałęzie drzewa, uważając na to, żeby nie stworzyć w ten sposób żadnej ścieżki dłuższej niż d. Do pierwszego wierzchołka na ścieżce nie możemy zatem już niczego doczepić, do drugiego możemy doklejać tylko pojedyncze krawędzie, do trzeciego – tylko poddrzewa o głębokości nie większej niż 2 itd. Maksymalne głębokości dołączanych poddrzew w przypadku $d = 5$ obrazuje poniższy rysunek.



Z tych ograniczeń wynika, że wszystkie ścieżki o długości d w drzewie muszą zawierać (co najmniej) jedną wspólną krawędź, a mianowicie tę środkową. Po jej usunięciu drzewo rozpada się na dwa mniejsze, które można ukorzenić w wierzchołkach z usuwanej krawędzi, otrzymując dwa drzewa o głębokości $\left\lfloor \frac{d}{2} \right\rfloor$. Jeżeli przez $u(m, g)$ oznaczymy liczbę nieizomorficznych ukorzenionych drzew m-wierzchołkowych o głębokości g, to wynikiem dla d nieparzystego będzie

$$\sum_{n_i=1}^{\lfloor (n-1)/2 \rfloor} \left(u\left(n_i, \left\lfloor \frac{d}{2} \right\rfloor\right) \cdot u\left(n - n_i, \left\lfloor \frac{d}{2} \right\rfloor\right) \right) + \\ + [2 \mid n] \cdot \binom{u\left(\frac{n}{2}, \left\lfloor \frac{d}{2} \right\rfloor\right) + 1}{2}.$$

Każdy ze składników w powyższej sumie odpowiada sklejeniu szukanego drzewa z dwóch poddrzew o głębokości $\left\lfloor \frac{d}{2} \right\rfloor$ zawierających odpowiednio n_i i $n - n_i$ wierzchołków, natomiast ostatni składnik (w którym $[2 \,|\, n]$, zapisane za pomocą tzw. notacji Iversona, jest równe 1, jeśli n jest parzyste, a 0 w przeciwnym przypadku) reprezentuje drzewo sklejone z dwóch poddrzew zawierających po $\frac{n}{2}$ wierzchołków. Zagadka dla Czytelnika: dlaczego ten ostatni przypadek musiał zostać potraktowany osobno? (Wskazówka: interesują nas tylko drzewa parami nieizomorficzne).

A co, jeżeli d jest parzyste? Wówczas środkowy wierzchołek na ścieżce wyznaczającej średnicę drzewa jest wspólny dla wszystkich takich ścieżek i jeżeli ukorzenimy w nim nasze drzewo, to otrzymamy n-wierzchołkowe drzewo o głębokości $\frac{d}{2}$. Stąd wynik to $u\left(n, \frac{d}{2}\right) \ldots$ no, prawie. Otóż trzeba jeszcze pamiętać o tym, że interesują nas tylko takie drzewa o głębokości $\frac{d}{2}$, w których co najmniej dwojgu dzieci korzenia odpowiadają poddrzewa o głębokości $\frac{d}{2} - 1$, stąd od $u\left(n, \frac{d}{2}\right)$ musimy odjąć liczbę drzew z tylko jednym takim dzieckiem, równą:

$$\sum_{n_i=1}^{n-1} u\left(n_i, \frac{d}{2} - 1\right) \cdot w\left(n - n_i, \frac{d}{2} - 1\right).$$

Przez $w(m, g)$ oznaczyliśmy liczbę ukorzenionych drzew m-wierzchołkowych o głębokości *nie większej niż* g (wyjaśnienie tego wzoru pozostawiam Czytelnikowi).

Pozostaje pytanie, jak obliczyć wartości $u(m, g)$ i $w(m, g)$. Ponieważ $u(m, g) = w(m, g) - w(m, g - 1)$, to skupimy się tylko na wartościach w. Zastosujemy programowanie dynamiczne, w którym będziemy wyznaczać wartości w dla kolejnych $g = 0, 1, 2, \ldots$ Dla $g = 0$ jest łatwo: $w(1, 0) = 1$, a dla $m > 1$ mamy $w(m, 0) = 0$. Jeżeli zaś $g > 0$, to drzewo o głębokości co najwyżej g otrzymujemy, podpinając krawędziami do korzenia pewną liczbę poddrzew o głębokości $g - 1$. Musimy być jednak ostrożni, aby w ten sposób nie obliczyć wielokrotnie żadnych drzew różniących się tylko kolejnością podpiętych poddrzew. W tym celu przy obliczaniu rządka $w(*, g)$ będziemy rozważać kolejno wartości $w(j, g - 1)$ dla $j = 1, 2, \ldots$; dla każdej z nich, przy aktualizacji danego pola $w(i, g)$ wybierzemy liczbę k poddrzew j-wierzchołkowych o głębokości co najwyżej $g - 1$, które zamierzamy podpiąć (oczywiście musi być $k \cdot j \leqslant i$). Liczba układów takich poddrzew będzie wówczas równa liczbie k-elementowych *kombinacji z powtórzeniami* ze zbioru $w(j, g - 1)$-elementowego, tzn.:

$$wsp(j, g - 1, k) = \binom{w(j, g - 1) + k - 1}{k}.$$

Ostatecznie otrzymujemy poniższy pseudokod, w którym zakładamy, że $w(*, -1) \equiv 0$ (pytania kontrolne: dlaczego pętla po j jest przed pętlą po i? i skąd nietypowy porządek przeglądania wartości w drugiej z tych pętli?).

for $g = 0, 1, 2, \ldots$
  $w(1, g) := 1$; $w(2, g) := w(3, g) := \ldots := w(n, g) := 0$;
  for $j = 1, 2, \ldots, n$
    for $i = n, n - 1, \ldots, 1$
      for $1 \leqslant k \leqslant \frac{i}{j}$
        $w(i, g) := w(i, g) + wsp(j, g - 1, k) \cdot w(i - kj, g)$;

Zastanówmy się na koniec nad złożonością czasową opisanego rozwiązania. Zauważmy przede wszystkim, że współczynniki wsp możemy obliczać w stałej złożoności czasowej, jeżeli tylko spamiętamy odwrotności liczb $1, \ldots, n + 1$ modulo p i będziemy każdorazowo korzystać ze wzoru:

$$wsp(j, g - 1, k + 1) = \\ = wsp(j, g - 1, k) \cdot (w(j, g - 1) + k) \cdot ((k + 1)^{-1} \bmod p).$$

To daje oszacowanie na złożoność czasową rozwiązania $O(dn^3)$. Okazuje się jednak, że jest ono zbyt pesymistyczne, gdyż łączna liczba obrotów trzech wewnętrznych pętli **for** może zostać oszacowana z góry przez

$$n^2 + \sum_{j=1}^{n} \sum_{i=1}^{n} \left\lfloor \frac{i}{j} \right\rfloor \leqslant n^2 + \sum_{i=1}^{n} i \cdot \sum_{j=1}^{n} \frac{1}{j} \leqslant n^2 + n^2 H_n,$$

gdzie H_n to tak zwana n-ta liczba harmoniczna. A ponieważ $H_n = O(\log n)$, co wynika m.in. z przybliżenia tej sumy całką, to otrzymujemy ostatecznie, że złożoność czasowa opisanego rozwiązania to tak naprawdę $O(dn^2 \log n)$.

Jakub RADOSZEWSKI