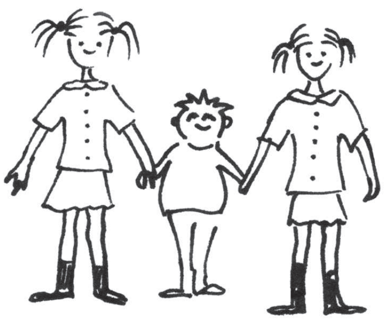
Rozwiązanie zadania M 1243.
Zauważmy, że jeśli liczba k należy do zbioru wartości funkcji f, to liczba $k+1$ też należy do zbioru wartości tej funkcji. Istotnie: jeśli $f(m)=k$ dla pewnego $m\in\mathbb{N}$, to $f(f(m))=f(m)+1=k+1$. Ponadto liczba 1 należy do zbioru wartości funkcji f. Stąd wynika, że zbiorem wartości funkcji f jest cały zbiór $\mathbb{N}$.

Wobec tego, jeśli $n\in\mathbb{N}$, to istnieje taka liczba $m\in\mathbb{N}$, że $f(m)=n$. Stąd $f(n)=f(f(m))=f(m)+1=n+1$. Otrzymaliśmy sprzeczność, gdyż wobec $f(n)\geqslant 2$ dla każdego $n\in\mathbb{N}$, wartość 1 nie może być przyjmowana.



Informatyczny kącik olimpijski (20): Palindromy

W tym kąciku informatycznym – zadanie *Palindromy* z finału XIII Olimpiady Informatycznej.

Mamy danych n różnych palindromów (czyli słów, które są takie same czytane od początku, jak i od końca) złożonych z małych liter alfabetu angielskiego (a więc mamy co najwyżej 26 różnych liter). Wiemy, że sumaryczna długość wszystkich palindromów jest nie większa niż m. Pytanie jest następujące: ile jest różnych par (uporządkowanych) podanych palindromów, które sklejone dają także palindrom?

Na początku wprowadźmy kilka oznaczeń. Jeśli A jest słowem, to przez $\overline{A}$ oznaczymy A czytane od tyłu. Zatem A jest palindromem, jeśli $A=\overline{A}$. Dalej, przez $A|_k$ oznaczymy prefiks A o długości k (a więc pierwsze k liter słowa A), za to przez $A|^k$ – sufiks A o długości k (tj. k ostatnich liter A). I jeszcze, niech $|A|$ będzie długością słowa A.

Zacznijmy od kilku spostrzeżeń, które narzucają się po przeanalizowaniu choćby przykładu do tego zadania – Czytelnikowi proponuję rozpisanie własnego i przyjrzenie mu się. Po pierwsze, na pewno każdy palindrom A z wejścia zwiększa wynik co najmniej o 1. Jest tak dlatego, że skoro $A=\overline{A}$, to $\overline{AA}=AA$. Po drugie, niech A i B będą takimi palindromami, że AB też jest palindromem, oraz niech $k=|A|<|B|=l$ (zachodzi albo to, albo nierówność w drugą stronę – długości nie mogą być sobie równe, bo wtedy mielibyśmy $A=B$). Wtedy BA również jest palindromem. Faktycznie, zachodzi $\overline{A}=A=\overline{B|^k}=B|_k$, a ponadto $B|_{l-k}$ jest palindromem, więc $B|^{l-k}$ również. Łatwo zauważyć, że występuje tutaj równoważność, czyli jeśli BA jest palindromem, to AB również.

Sumarycznym wynikiem jest więc $n+2q$, gdzie q to liczba takich par różnych palindromów AB, że $|A|<|B|$.

No to teraz możemy zastanowić się nad tym, kiedy dwa palindromy A i B ($|A|<|B|$) tworzą palindrom AB. Na pewno trzeba, żeby $A=\overline{B|^{|A|}}=B|_{|A|}$, czyli żeby A było prefiksem B. Poza tym jeszcze $B|_{|B|-|A|}$ musi być palindromem (jest to środek AB, po ucięciu z początku i z końca $|A|$ liter). Na bazie tych dwóch warunków zbudujemy już rozwiązanie zadania.

Zacznijmy od tego, jak znaleźć wszystkie palindromy A z wejścia, będące prefiksami palindromu B? Załóżmy, że wejście jest posortowane leksykograficznie. Zauważmy, że jeśli A jest prefiksem B, to w kolejności leksykograficznej między nimi mogą znaleźć się tylko takie słowa C, że A jest prefiksem C. Załóżmy, że mamy policzoną listę L wszystkich spośród naszych palindromów, które są prefiksami C, C występuje bezpośrednio przed B w kolejności leksykograficznej oraz znamy maksymalne możliwe i spełniające $C|_i=B|_i$ (obliczenie takiego i to koszt $O(|B|+|C|)$). Wtedy lista wszystkich spośród naszych słów, które są prefiksami B, to lista L uzupełniona o C, z której zostały usunięte wszystkie słowa dłuższe niż i. Długość tej listy dla C jest ograniczona przez długość C (na wejściu mamy parami różne palindromy), a więc cała operacja otrzymania listy dla B z listy dla C ma koszt $O(|B|+|C|)$.

W drugim kroku chcielibyśmy sprawdzić, dla danej pary palindromów A i B, gdzie A występuje na liście odpowiadającej B, czy rzeczywiście AB jest palindromem, czyli czy $B|_{|B|-|A|}$ też nim jest. $B|_i$ jest palindromem wtedy i tylko wtedy, gdy $B|_i=\overline{B|^i}=B|^i$. A w takim razie $B|_i$ musi być jednocześnie i prefiksem, i sufiksem słowa B. A to już brzmi znajomo – w tym momencie warto skorzystać z algorytmu Knutha–Morrisa–Pratta (Czytelnikowi nieobeznanemu z tym algorytmem polecam skorzystanie z mnóstwa materiałów dostępnych w Internecie). W tym algorytmie wyznaczana jest tablica prefiksowa P, w której $P[i]$ jest długością najdłuższego prefiksu, będącego jednocześnie sufiksem słowa $B|_i$. Zauważmy, że długością najdłuższego prefiksu, będącego jednocześnie sufiksem B jest $P[|B|]$, następnego najdłuższego – $P[P[|B|]]$, itd., aż w końcu $P[P[\ldots[P[|B|]]\ldots]]=0$. Algorytm KMP oblicza tablicę P w czasie $O(|B|)$, a my w takim samym czasie możemy odczytać z niej długości wszystkich prefiksów B będących sufiksami B, czyli palindromami. Po tych wstępnych obliczeniach można już natychmiastowo sprawdzić, dla każdego palindromu A z listy odpowiadającej B, czy $B|_{|B|-|A|}$ jest palindromem.

W takim razie sumaryczny czas, jaki musimy poświęcić B, jest rzędu $O(|B|+|C|)$, gdzie C jest poprzednim leksykograficznie palindromem, a więc sumaryczny czas całego algorytmu, nie licząc sortowania, to $O(m)$. Czytelnikowi pozostawiam sprytną implementację algorytmu sortowania pozycyjnego tak, żeby złożoność całego algorytmu pozostała $O(m)$. Liczba „obrotów” sortowania może być bliska $26m$, ze względu na rozmiar alfabetu – chociaż tę stałą da się znacząco zmniejszyć, co również polecam jako ćwiczenie.

Filip WOLSKI