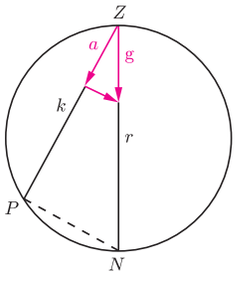
Rozwiązanie zadania M 1252.
Dla każdej liczby rzeczywistej x spełniona jest nierówność $x(1-x) \leqslant \frac{1}{4}$. Wobec tego

$$a(1-b)\cdot b(1-c)\cdot c(1-a) =$$
$$= a(1-a)\cdot b(1-b)\cdot c(1-c) \leqslant \left(\frac{1}{4}\right)^3,$$

skąd wynika, że co najmniej jedna z liczb $a(1-b)$, $b(1-c)$, $c(1-a)$ nie przekracza $1/4$.

Rozwiązanie zadania F 747.
Niech r będzie liczbą większą od k i od l. W pionowym kole o promieniu $r/2$ rysujemy z jego najwyższego punktu cięciwę o długości k.



Ruch kulki staczającej się po deseczce umieszczonej tak, jak ta cięciwa, trwa tyle samo czasu (oznaczmy go przez t_1) co swobodny spadek wzdłuż pionowej średnicy koła (ten czas oznaczmy przez t). Jest tak dlatego, że z podobieństwa trójkątów mamy

$$\frac{a}{g} = \frac{ZP}{ZN} = \frac{at_1^2/2}{gt^2/2},$$

skąd po skróceniu wynika, że $t_1^2 = t^2$, a ponieważ obie liczby są dodatnie, więc $t_1 = t$.

Umieszczenie w tym samym kole cięciwy o długości l zaczynającej się w najwyższym punkcie daje rozwiązanie.

Ze względu na dowolność wyboru r zadanie ma nieskończenie wiele rozwiązań.

Informatyczny kącik olimpijski (23): Haszowanie

Tym razem przyjrzymy się zadaniu *Haszowanie* z Obozu Naukowo-Treningowego im. A. Kreczmara 2008.

Niech L będzie zbiorem słów o długości co najwyżej 20 złożonych z małych liter alfabetu angielskiego, a $\mathbb{Z}_n$ – zbiorem reszt z dzielenia przez n. Mając dane $n < 10^{16}$ i pewną funkcję $f : L \to \mathbb{Z}_n$, mamy znaleźć takie dwa słowa $u, v \in L$, że $f(u) = f(v)$. Pisząc, że funkcja f jest „dana”, mam na myśli, że możemy spytać o jej wartości dla konkretnych słów, które nas interesują. Chodzi właśnie o znalezienie rozwiązania zadania, korzystającego z możliwie niewielkiej liczby takich zapytań.

Zauważmy najpierw, że wystarczy wziąć dowolne $n+1$ słów (z warunków zadania wynika, że $n < |L|$) i w tym zbiorze na pewno znajdą się dwa spełniające warunki zadania. Daje to algorytm wykonujący $O(n)$ zapytań, wykorzystujący pamięć $O(n)$.

Okazuje się, że przy losowym wyborze słów ze zbioru L, oczekiwana liczba zapytań (i zużycie pamięci) przy opisanej metodzie to $O(\sqrt{n})$. Proponuję Czytelnikowi zastanowienie się, dlaczego tak jest (jest to związane z tzw. *paradoksem urodzin*) – rozwiązanie znajduje się w innym miejscu numeru.

Zapamiętanie $\sqrt{n}$ słów o długości rzędu 20 znaków może być jednak przy podanych ograniczeniach dość kłopotliwe. Omówimy teraz rozwiązanie niniejszego zadania działające w stałej pamięci i oczekiwanej złożoności czasowej $O(\sqrt{n})$.

Obierzmy funkcję różnowartościową $g : \mathbb{Z}_n \to L$, o pseudolosowych wartościach, której wartość dla konkretnej liczby z $\mathbb{Z}_n$ będziemy umieli łatwo obliczyć. Dobrym przykładem jest funkcja zwracająca w punkcie x słowo, które jest w L leksykograficznie na $(x \oplus 1357)$ miejscu ($\oplus$ to różnica symetryczna na bitach – *xor*).

Zdefiniujmy $h : \mathbb{Z}_n \to \mathbb{Z}_n$, tak żeby $h(x) = f(g(x))$. Problem sprowadza się do znalezienia takich różnych x i y, że $h(x) = h(y)$, wówczas bowiem $g(x)$ i $g(y)$ będą naszymi szukanymi słowami. Teoretycznie, h mogłoby być permutacją zbioru $\mathbb{Z}_n$, a liczby x i y mogłyby w ogóle nie istnieć, jest to jednak zupełnie nieprawdopodobne przy losowym wyborze g.

Jak znaleźć takie x i y w stałej pamięci? Obierzmy w tym celu losowe $p \in \mathbb{Z}_n$ i rozważmy ciąg

$$(a_i) = (p, h(p), h(h(p)), h(h(h(p))), \ldots, h^i(p), \ldots).$$

Możemy przyjąć, że $(g(a_i))$ jest losowym ciągiem słów (bo funkcja g zwraca losowe słowa), czyli wśród jego pierwszych $O(\sqrt{n})$ wyrazów z dużym prawdopodobieństwem znajdują się dwa o równych obrazach przy przekształceniu f. Na odpowiadających im wyrazach ciągu (a_i) funkcja h także ma jednakowe wartości, a stąd także w samym (a_i) z prawdopodobieństwem bliskim 1 występuje powtórzenie po $O(\sqrt{n})$ wyrazach. Niech pierwszym elementem, który się powtarza, będzie a_i, które stoi także na pozycji $j+1$. Ciąg (a_i) wygląda więc następująco:

$$a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_j, a_i, a_{i+1}, \ldots, a_j, a_i, a_{i+1}, \ldots,$$

gdzie j jest rzędu $\sqrt{n}$, a a_{i-1} i a_j to szukana para liczb. Żeby je odszukać, należy porównywać, dla kolejnych $k = 1, 2, \ldots$, wyraz a_k z a_{2k} oraz a_{2k+1}. Gdy $k = i$, wszystkie te trzy wyrazy należą do cyklu $a_i, \ldots, a_j$, a przez kolejne $j - i + 1$ kroków a_k zatacza pełen cykl, a dwa pozostałe zataczają go nawet dwukrotnie. Po drodze któryś z tych dwóch musi się więc spotkać z a_k. Niech takim spotkaniem będzie $a_k = a_l$. Wtedy $l - k$ jest wielokrotnością długości naszego cyklu (czyli $l - k = c(j - i + 1)$ dla pewnego $c \in \mathbb{Z}_+$). Biorąc p i $h^{l-k}(p)$ i aplikując do nich h tak długo, jak długo są różne, otrzymamy żądane a_{i-1} i a_j. W każdym z dwóch przeszukiwań wykonujemy $O(j)$ kroków, a więc oczekiwana liczba wywołań funkcji f jest równa $O(\sqrt{n})$.

Tomasz KULCZYŃSKI