Krzysztof Sikora
Treść zadania, Opracowanie

Paweł Parys
Program

OI, Etap III, dzień próbny, 28.03.2006

Tańce w kółkach

Do pewnego przedszkola chodzi n dzieci, które codziennie ustawiają się w k kółek i tańczą. W każdym kółku tańczy co najmniej l dzieci. Dwa ustawienia dzieci uważamy za różne, jeżeli pewne dziecko w jednym ustawieniu ma innego sąsiada po swojej prawej stronie niż w drugim.

Twoim zadaniem jest obliczenie liczby wszystkich różnych ustawień modulo 2005. Jeżeli nie ma ustawień spełniających opisane warunki, poprawnym wynikiem jest 0.

Zadanie

Napisz program, który:

- *wczyta ze standardowego wejścia liczby n, k oraz l,*
- *obliczy liczbę $d' = d \bmod 2005$, gdzie d jest liczbą różnych ustawień dzieci ($d \bmod 2005$ oznacza resztę z dzielenia d przez 2005),*
- *wypisze d' na standardowe wyjście.*

Wejście

Pierwszy i jedyny wiersz standardowego wejścia zawiera trzy liczby całkowite oddzielone pojedynczymi odstępami: n — liczba dzieci ($3 \leqslant n \leqslant 1\,000\,000\,000$), k — liczba kółek ($1 \leqslant k \leqslant n$) oraz l — minimalna liczba dzieci w każdym kółku ($2 \leqslant l \leqslant n$).

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę: $d \bmod 2005$.

Przykład

Dla danych wejściowych:

```
7 2 3
```

poprawnym wynikiem jest:

```
420
```

Rozwiązanie

Rozwiązanie podstawowe

Niech $S(n,k,l)$ oznacza liczbę ustawień n dzieci w k kółkach, w każdym przynajmniej l dzieci. Dla $n \geqslant k \cdot l$ oraz $k > 0$ mamy własność:

$$S(n,k,l) = (n-1) \cdot S(n-1,k,l) + (n-1) \cdots (n-l+1) \cdot S(n-l,k-1,l) \qquad (1)$$

Można ją wytłumaczyć w następujący kombinatoryczny sposób. Spójrzmy na pewne ustalone dziecko d. Wtedy wszystkie ustawienia dzieci możemy podzielić na dwie rozłączne grupy, zależnie od rozmiaru kółka, w którym stoi dziecko d:

- ustawienia, w których dziecko d znajduje się w kółku o rozmiarze większym niż l. Pozostałe $n-1$ dzieci można ustawić w kółkach na $S(n-1,k,l)$ sposobów. W każdym takim układzie możemy umieścić dziecko d na prawo od dowolnego z pozostałych $n-1$ dzieci, uzyskując ustawienie, w którym dziecko d występuje w kółku o rozmiarze przynajmniej $l+1$. To daje nam $(n-1) \cdot S(n-1,k,l)$ układów — pierwszy składnik sumy,

- ustawienia, w których dziecko d znajduje się w kółku o rozmiarze l. Pozostałe $l-1$ dzieci do tego kółka możemy wybrać na $\binom{n-1}{l-1}$ sposobów i ustawić je w kółko na $(l-1)!$ sposobów. Pozostałe $n-l$ dzieci mogą być ustawione na $S(n-l,k-1,l)$ sposobów. To daje nam $(n-1) \cdots (n-l+1) \cdot S(n-l,k-1,l)$ układów — drugi składnik sumy.

Dla $n < k \cdot l$ oraz dla $k = 0$ i $n > 0$ mamy $S(n,k,l) = 0$. Ponadto $S(0,0,l) = 1$. Wykorzystując równanie (1) i programowanie dynamiczne możemy łatwo rozwiązać zadanie. Na początku obliczamy iloczyny $(i-1) \cdots (i-l+1)$ dla każdego $i \leqslant n$. Następnie liczymy $S(i,j,l)$ dla wszystkich $i \leqslant n$, $j \leqslant k$, przy czym podczas wyznaczania wartości funkcji S dla i i j wykorzystujemy wcześniej policzone wartości funkcji S dla mniejszych argumentów. Wszystkie działania wykonujemy modulo 2005 stosując dobrze znane własności kongruencji stanowiące, iż jeśli

$$a \equiv b \pmod p \quad \text{oraz} \quad c \equiv d \pmod p$$

to

$$a+c \equiv b+d \pmod p \quad \text{oraz} \quad a \cdot c \equiv b \cdot d \pmod p.$$

Przedstawiony algorytm ma złożoność czasową $O(n \cdot (k+l))$, a pamięciową $O(n)$. Nieźle, jednak wystarczy to do zaliczenia tylko około połowy testów. Dla rozwiązania problemu dla większych n potrzebny jest szybszy program.

Rozwiązanie wzorcowe

Rozwiązanie podstawowe można przyspieszyć spostrzegając, że wśród wartości funkcji S modulo 2005 występuje wiele zer. Aby je skutecznie wyznaczyć wprowadzimy dodatkowe funkcje.

Alternatywne sformułowanie S

Dla nieujemnych liczb całkowitych $c_l, c_{l+1}, \ldots, c_n$ niech

$$A(c_l, c_{l+1}, \ldots, c_n) = c_l! \cdot l^{c_l} \cdot c_{l+1}! \cdot (l+1)^{c_{l+1}} \cdot \ldots \cdot c_n! \cdot n^{c_n}$$

oraz niech

$$B(c_l, c_{l+1}, \ldots, c_n) = \frac{n!}{A(c_l, c_{l+1}, \ldots, c_n)}$$

Wówczas zachodzi następująca równość:

$$S(n,k,l) = \sum B(c_l, c_{l+1}, \ldots, c_n), \tag{2}$$

gdzie suma przebiega po wszystkich ciągach $c_l, c_{l+1}, \ldots, c_n$ takich, że $\sum_{i=l}^{n} c_i = k$ oraz $\sum_{i=l}^{n} i \cdot c_i = n$.

Aby uzasadnić poprawność powyższej zależności przyjrzyjmy się jej bliżej. Zauważmy, że warunki nałożone na wartości $c_l, c_{l+1}, \ldots, c_n$ pozwalają nam interpretować c_i jako liczbę kółek rozmiaru i. Tak więc każdy ciąg $c_l, c_{l+1}, \ldots, c_n$ odpowiada konkretnej konfiguracji rozmiarów kółek, stąd odpowiadający mu składnik sumy (2) powinien oznaczać liczbę możliwych ustawień dzieci w tej konfiguracji. Upewnijmy się, że właściwie wyznaczyliśmy tę liczbę — obliczmy, na ile sposobów można rozmieścić dzieci w kółkach o podanych rozmiarach. Rozważmy następujący sposób formowania kółek: ustawmy wszystkie dzieci w szeregu (możemy to zrobić na $n!$ sposobów), a następnie podzielmy je kolejno na grupy: c_l grup po l dzieci, c_{l+1} grup po $l+1$ dzieci itd. — na koniec z każdej grupy utwórzmy kółko. Oczywiście przy takim podejściu każdą konfigurację tańca w kółkach o zadanych rozmiarach $c_l, c_{l+1}, \ldots, c_n$ możemy otrzymać wiele razy. Jak wiele? Po pierwsze w każdym kółku nie ma znaczenia, które dziecko jest pierwsze, więc każde ustawienie dzieci w kółkach liczymy $l^{c_l} \cdot (l+1)^{c_{l+1}} \cdot \ldots \cdot n^{c_n}$ razy. Dodatkowo także kolejność kółek tego samego rozmiaru nie ma znaczenia, więc musimy domnożyć tę liczbę przez $c_l! \cdot c_{l+1}! \cdot \ldots \cdot c_n!$. Ostatecznie otrzymujemy, że każde ustawienie powtarza się $A(c_l, c_{l+1}, \ldots, c_n)$ razy wśród wszystkich $n!$ szeregów, tak więc mamy $B(c_l, c_{l+1}, \ldots, c_n)$ różnych ustawień dla danych rozmiarów kółek.

Niezerowe konfiguracje

Postaramy się teraz odpowiedzieć na pytanie, kiedy dla liczby pierwszej p zachodzi nierówność $B(c_l, \ldots, c_n) \not\equiv 0 \pmod p$? Niech $X(m,p)$ będzie największym i takim, że $p^i | m$. Zauważmy, że $B(c_l, \ldots, c_n) \not\equiv 0 \pmod p$ tylko wtedy, gdy wszystkie potęgi p z licznika i z mianownika skracają się, a tak dzieje się tylko wówczas, gdy $X(A(c_l, \ldots, c_n), p) = X(n!, p)$. Wiemy, że

$$X(n!,p) = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \cdots$$

Spróbujmy także oszacować $X(A(c_l, \ldots, c_n), p)$. W tym celu zauważmy, że:

$$\begin{aligned}
X(A(c_l, \ldots, c_n), p) &= X(c_l! \cdot l^{c_l} \cdot c_{l+1}! \cdot (l+1)^{c_{l+1}} \cdot \ldots \cdot c_n! \cdot n^{c_n}, p) \\
&= X(c_l! \cdot l^{c_l} \cdot c_{l+1}! \cdot (l+1)^{c_{l+1}} \cdot \ldots \cdot c_{p-1}! \cdot (p-1)^{c_{p-1}}, p) \\
&+ X(c_p! \cdot p^{c_p}, p) + \cdots + X(c_n! \cdot n^{c_n}, p)
\end{aligned}$$

Rozważymy odrębnie poszczególne składniki tej sumy.

Przypadek 1. Dla kółek rozmiaru $i = p$ mamy:

$$\begin{aligned} X(c_p! \cdot p^{c_p}, p) &= X(c_p!, p) + X(p^{c_p}, p) \\ &= c_p + \left\lfloor \frac{c_p}{p} \right\rfloor + \left\lfloor \frac{c_p}{p^2} \right\rfloor + \cdots \\ &= \left\lfloor \frac{c_p p}{p} \right\rfloor + \left\lfloor \frac{c_p p}{p^2} \right\rfloor + \left\lfloor \frac{c_p p}{p^3} \right\rfloor + \cdots \\ &= X((c_p p)!, p). \end{aligned} \tag{3}$$

Przypadek 2. Teraz przyjrzyjmy się kółkom o rozmiarze $i > p$. Jeśli $p \geqslant 3, j \geqslant 2$, to zachodzi nierówność $1 + j \leqslant p^{j-1}$. To daje nam $(1 + X(i,p)) \cdot p \leqslant i$ dla wszystkich $p \geqslant 3, i > p$ (dla $j = X(i,p) \geqslant 2$ działa podana nierówność, a pozostałe przypadki łatwo sprawdzić). Otrzymujemy wtedy:

$$\begin{aligned} X(c_i! \cdot i^{c_i}, p) &\leqslant c_i + X(c_i! \cdot i^{c_i}, p) \\ &= c_i + X(i,p) \cdot c_i + \left\lfloor \frac{c_i}{p} \right\rfloor + \left\lfloor \frac{c_i}{p^2} \right\rfloor + \cdots \\ &= \left\lfloor \frac{(1 + X(i,p))c_i \cdot p}{p} \right\rfloor + \left\lfloor \frac{c_i \cdot p}{p^2} \right\rfloor + \left\lfloor \frac{c_i \cdot p}{p^3} \right\rfloor + \cdots \\ &\leqslant \left\lfloor \frac{c_i i}{p} \right\rfloor + \left\lfloor \frac{c_i i}{p^2} \right\rfloor + \left\lfloor \frac{c_i i}{p^3} \right\rfloor + \cdots \\ &= X((c_i i)!, p). \end{aligned} \tag{4}$$

Przy czym równość może zachodzić jedynie wtedy, gdy $c_i = 0$.

Przypadek 3. Rozważmy łącznie pozostałe kółka rozmiaru mniejszego niż p. Załóżmy, że takie kółka mogą istnieć, czyli $l < p$. Niech $m = c_l l + c_{l+1}(l+1) + \ldots + c_{p-1}(p-1)$. Ponieważ $l \geqslant 2$, to zachodzi nierówność $2 \cdot (c_l + \cdots + c_{p-1}) \leqslant m$. Wstępnie zauważmy, że

$$X(c_l! \cdot l^{c_l} \cdot \ldots \cdot c_{p-1}! \cdot (p-1)^{c_{p-1}}, p) = X(c_l! \cdot \ldots \cdot c_{p-1}!, p).$$

Dalej możemy przeprowadzić następujące szacowanie:

$$\begin{aligned} X(c_l! \cdots c_{p-1}!, p) &\leqslant \left\lfloor \frac{m}{p} \right\rfloor - \left\lfloor \frac{m}{2p} \right\rfloor + X(c_l! \cdots c_{p-1}!, p) \\ &= \left\lfloor \frac{m}{p} \right\rfloor - \left\lfloor \frac{m}{2p} \right\rfloor + \left\lfloor \frac{c_l}{p} \right\rfloor + \cdots + \left\lfloor \frac{c_{p-1}}{p} \right\rfloor + \left\lfloor \frac{c_l}{p^2} \right\rfloor + \cdots + \left\lfloor \frac{c_{p-1}}{p^2} \right\rfloor + \cdots \\ &\leqslant \left\lfloor \frac{m}{p} \right\rfloor - \left\lfloor \frac{m}{2p} \right\rfloor + \left\lfloor \frac{c_l + \cdots + c_{p-1}}{p} \right\rfloor + \left\lfloor \frac{c_l + \cdots + c_{p-1}}{p^2} \right\rfloor + \cdots \\ &\leqslant \left\lfloor \frac{m}{p} \right\rfloor - \left\lfloor \frac{m}{2p} \right\rfloor + \left\lfloor \frac{m}{2p} \right\rfloor + \left\lfloor \frac{m}{p^2} \right\rfloor + \left\lfloor \frac{m}{p^3} \right\rfloor + \cdots \\ &= X(m!, p). \end{aligned} \tag{5}$$

Zauważmy, że dla $m < p$ zachodzi $\left\lfloor \frac{m}{p} \right\rfloor = \left\lfloor \frac{m}{2p} \right\rfloor = 0$, a w przeciwnym przypadku mamy $\left\lfloor \frac{m}{p} \right\rfloor > \left\lfloor \frac{m}{2p} \right\rfloor$ i wtedy na pewno w powyższym ciągu porównań nie zachodzi równość.

Podsumowując wszystkie trzy przypadki dostajemy oszacowanie:

$$\begin{aligned} X(A(c_l,\ldots,c_n),p) &\leqslant X(m!,p)+X((c_p\cdot p)!,p)+\ldots+X((c_n\cdot n)!,p) \\ &= X(m!\cdot(c_p\cdot p)!\cdot\ldots\cdot(c_n\cdot n)!,p) \end{aligned}$$

Dodatkowo łatwo zauważyć, że prawdziwa jest następująca prosta własność $X(n_1!\cdots n_s!,p)\leqslant X((n_1+\cdots+n_s)!,p)$, więc mamy

$$\begin{aligned} X(A(c_l,\ldots,c_n),p) &\leqslant X((m+c_p\cdot p+\ldots+c_n\cdot n)!,p) \\ &= X((c_l\cdot l+\ldots+c_n\cdot n)!,p) \\ &= X(n!,p) \end{aligned}$$

Co więcej, z przeprowadzonych obliczeń wiemy, że powyższa nierówność może być równością tylko wtedy, gdy $m<p$ oraz $c_i=0$ dla $i>p$. Otrzymujemy stąd następujący wniosek:

Wniosek 1 (O niezerowości) *Niech m będzie liczbą dzieci w kółkach mniejszych niż p, to znaczy $m=c_l l+c_{l+1}(l+1)+\cdots+c_{p-1}(p-1)$. Jeśli $B(c_l,\ldots,c_n)\not\equiv 0 \pmod p$, to $m<p$ oraz $c_i=0$ dla każdego $i>p$.*

Uproszczona rekurencja

Z wniosku 1 wynika, że poszukując niezerowych wartości $B(c_l,\ldots,c_n)\not\equiv 0 \pmod p$ wystarczy rozważyć konfiguracje, w których nie ma kółek rozmiaru większego niż p i mniej niż p dzieci łącznie stoi w kółkach rozmiaru mniejszego niż p. To oznacza, że musimy sformować $c_p=\lfloor n/p\rfloor$ kółek rozmiaru p, a pozostałe $n \bmod p$ dzieci rozmieścić w kółkach rozmiaru mniejszego niż p. Widzimy więc, że zachodzi następujący wzór:

$$\begin{aligned} S(n,k,l) &\equiv \binom{n}{c_p\cdot p}\cdot\frac{(c_p\cdot p)!}{c_p!\cdot p^{c_p}}\cdot S(n \bmod p,k-c_p,l) \pmod p \\ &\equiv \frac{n!}{c_p!\cdot p^{c_p}\cdot(n \bmod p)!}\cdot S(n \bmod p,k-c_p,l) \pmod p, \end{aligned}$$

Pierwszy czynnik drugiego wiersza możemy jeszcze bardziej uprościć. Powołujemy się przy tym na następujące ogólnie znane twierdzenie:

Twierdzenie 2 (Wilson) *Dla dowolnej liczby pierwszej p prawdą jest, że*

$$(p-1)!\equiv -1 \pmod p$$

To oznacza, że dla każdego j mamy $\prod_{i=1}^{p-1}(j\cdot p+i)\equiv -1 \pmod p$. Stąd odpowiednio grupując czynniki $n!$ otrzymujemy:

$$n!=\underbrace{\left(\prod_{k=0}^{c_p-1}\prod_{i=1}^{p-1}(k\cdot p+i)\right)}_{(*)}\cdot\underbrace{p^{c_p}\cdot c_p!}_{(**)}\cdot\underbrace{\left(\prod_{k=1}^{n \bmod p}(c_p\cdot p+k)\right)}_{(***)}.$$

Zauważmy, że:

(*) równa się $(-1)^{c_p}$ z twierdzenia Wilsona,

(**) skraca się z mianownikiem,

(***) podzielona przez $(n \bmod p)!$ równa się 1, co wynika stąd, iż:

$$(c_p \cdot p+1)\cdots(c_p \cdot p+n \bmod p) \equiv (n \bmod p)! \pmod{p} \tag{6}$$

i dalej, korzystając z faktu, że gdy

$$da \equiv db \pmod{c} \quad \text{oraz} \quad \mathrm{NWD}(d,c)=1$$

to

$$a \equiv b \pmod{c},$$

po podzieleniu kongruencji (6) stronami przez $(n \bmod p)!$ otrzymujemy:

$$\frac{(c_p \cdot p+1)\cdot(c_p \cdot p+2)\cdots(c_p \cdot p+n \bmod p)}{(n \bmod p)!} \equiv 1 \pmod{p}.$$

Stąd ostatecznie otrzymujemy

$$\frac{n!}{c_p! \cdot p^{c_p} \cdot (n \bmod p)!} \equiv (-1)^{c_p} \pmod{p}$$

i możemy sformułować:

Wniosek 3 (Uproszczona rekurencja)

$$S(n,k,l) \equiv (-1)^{\lfloor n/k \rfloor} \cdot S(n \bmod p, k-\lfloor n/k \rfloor, l) \pmod{p}$$

Algorytm

Wykorzystując uproszczoną rekurencję możemy rozwiązać zadanie dużo szybciej niż poprzednimi metodami. Dostajemy algorytm działający w czasie $O(p^2)$, gdzie p jest liczbą pierwszą, modulo która chcemy otrzymać wynik. Co prawda 2005 nie jest liczbą pierwszą, gdyż rozkłada się na czynniki $2005 = 5 \cdot 401$. Zauważmy jednak, że 5 i 401 to liczby pierwsze i możemy obliczyć liczbę różnych ustawień dzieci modulo te liczby — oznaczmy je odpowiednio przez a i b. Chińskie twierdzenie o resztach mówi, że w zbiorze $\{0,1,\ldots,2004\}$ jest dokładnie jedna liczba c, taka że $c \equiv a \pmod{5}$ i $c \equiv b \pmod{401}$. Najprostszy sposób, aby ją znaleźć, to sprawdzić wszystkich kandydatów od 0 do 2004.

Testy

Zadanie testowane było na zestawie 20 danych testowych pogrupowanych w pary.

Nazwa	**n**	**k**	**l**	**Opis**
tan1a.in	82	3	18	test poprawnościowy
tan1b.in	1000	1000	1000	test poprawnościowy
tan2a.in	400	200	2	test poprawnościowy
tan2b.in	1 000	22	10	test poprawnościowy
tan3a.in	976	6	43	test poprawnościowy
tan3b.in	988	3	200	test poprawnościowy
tan4a.in	4	1	3	test poprawnościowy
tan4b.in	800	21	21	test poprawnościowy
tan5a.in	167	5	18	test poprawnościowy
tan5b.in	477	18	5	test poprawnościowy
tan6a.in	5 555	1 111	2	mały test wydajnościowy
tan6b.in	197 512	144	3	mały test wydajnościowy
tan7a.in	50 000	130	9	mały test wydajnościowy
tan7b.in	20 000	2 000	4	średni test wydajnościowy
tan8a.in	1 234 581	3 079	116	duży test wydajnościowy
tan8b.in	1 000 000 000	500 111 000	102	duży test wydajnościowy
tan9a.in	999 999 492	2 493 765	105	duży test wydajnościowy
tan9b.in	1 000 000 000	10 000	8	duży test wydajnościowy
tan10a.in	1 000 000 000	200 000 000	4	duży test wydajnościowy
tan10b.in	999 999 999	5 988 023	76	duży test wydajnościowy