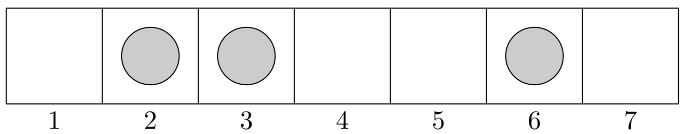
Bartosz Walczak — Treść zadania, Opracowanie
Jakub Pawlewicz — Programy

Gra

Rozważmy grę na prostokątnej planszy $m \times 1$ złożonej z m jednostkowych kwadratów ponumerowanych kolejno od 1 do m. Na planszy ustawionych jest n pionków, każdy na innym polu, przy czym żaden pionek nie znajduje się na polu o numerze m. Pojedynczy ruch w grze polega na przestawieniu dowolnie wybranego pionka na pierwsze wolne pole o większym numerze. Dwaj gracze wykonują na zmianę po jednym ruchu. Wygrywa ten, który postawi pionek na ostatnim polu, tzn. na polu o numerze m.



Dla przykładu z rysunku ($m = 7$), gracz może wykonać ruch z pola 2 na 4, z pola 3 na 4 lub z pola 6 na 7. Ten ostatni ruch kończy grę.

Mówimy, że ruch gracza jest wygrywający, jeżeli po jego wykonaniu gracz ten może wygrać grę niezależnie od tego, jakie ruchy będzie wykonywał jego przeciwnik.

Zadanie

Napisz program, który:

- *wczyta ze standardowego wejścia rozmiar planszy i początkowe rozstawienie pionków,*
- *wyznaczy liczbę różnych ruchów wygrywających, jakie w zadanej sytuacji początkowej ma do wyboru gracz rozpoczynający grę,*
- *wypisze wynik na standardowe wyjście.*

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite m i n ($2 \leqslant m \leqslant 10^9$, $1 \leqslant n \leqslant 10^6$, $n < m$) oddzielone pojedynczym odstępem. Drugi wiersz zawiera n rosnących numerów pól, na których znajdują się pionki. Liczby w wierszu są pooddzielane pojedynczymi odstępami.

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać liczbę różnych ruchów wygrywających, jakie może wykonać w zadanej sytuacji początkowej gracz rozpoczynający grę.

Przykład

Dla danych wejściowych:

```
5 2
1 3
```

prawidłową odpowiedzią jest:

```
1
```

Dla danych wejściowych:

```
5 2
2 3
```

prawidłową odpowiedzią jest:

```
0
```

Rozwiązanie

Wprowadzenie

Na początek wprowadzimy kilka pojęć, które ułatwią nam dalszą analizę problemu. I tak *pozycją początkową* będziemy nazywać pozycję, w której rozpoczyna się rozgrywka, natomiast *pozycją końcową* będziemy nazywać pozycję, w której nie można wykonać żadnego ruchu, czyli w której rozgrywka się kończy. Będziemy rozpatrywać tylko takie gry, w których każda rozgrywka jest skończona, czyli prowadzi do jakiejś pozycji końcowej.

Wszystkie możliwe rozgrywki możemy przedstawić przy pomocy tzw. *drzewa gry*. W takim drzewie będziemy utożsamiać wierzchołki z pozycjami osiągalnymi z pozycji początkowej, a krawędzie z ruchami możliwymi w danych pozycjach. Korzeń drzewa odpowiada pozycji początkowej, a liście — pozycjom końcowym.

Pojęcie *pozycji wygrywającej* i *przegrywającej* definiujemy przez indukcję w górę drzewa gry:

1. Każda pozycja końcowa jest przegrywająca.
2. Jeżeli w danej pozycji istnieje ruch prowadzący do pozycji przegrywającej, to pozycja jest wygrywająca.
3. Jeżeli w danej pozycji wszystkie ruchy prowadzą do pozycji wygrywających, to pozycja jest przegrywająca.

Warunek 1 wynika z warunku 3, ale został dodany dla zwiększenia czytelności.

Według powyższej definicji każda pozycja zostaje jednoznacznie zakwalifikowana jako wygrywająca, albo przegrywająca. Takie zakwalifikowanie pozycji będziemy określać mianem *strategii wygrywającej*. Wprowadzamy również pojęcie *ruchu wygrywającego* i *przegrywającego* jako ruchu prowadzącego, odpowiednio, do pozycji przegrywającej lub wygrywającej. To co otrzymujemy jest zgodne z intuicją: w pozycji wygrywającej istnieje ruch wygrywający, w pozycji przegrywającej wszystkie ruchy są przegrywające. Ponadto pojęcie ruchu wygrywającego z treści zadania jest równoważne tak zdefiniowanemu pojęciu ruchu wygrywającego.

W razie potrzeby możemy modyfikować drzewo gry, otrzymując drzewo innej gry, ze zmienionymi zasadami. Jak łatwo zauważyć, usunięcie z drzewa gry jakiejś krawędzi (i w efekcie całego poddrzewa pod tą krawędzią) odpowiadającej ruchowi przegrywającemu nie zmienia strategii wygrywającej. Jest to zgodne z intuicją, gdyż ruchów przegrywających nigdy nie opłaca się wykonywać.

Większość zadań o grach sprowadza się do znalezienia strategii wygrywającej. Tak jest również w tym przypadku. Oczywiście brutalne rozwiązanie polegające na przeszukiwaniu całego drzewa gry nie ma szans powodzenia już dla średniej wielkości danych, dlatego potrzebujemy łatwiej obliczalnego kryterium pozwalającego określić strategię wygrywającą. W jego znalezieniu pomoże nam analiza gier: Nim i „Staircase Nim".

Gra Nim

Planszą do gry Nim jest pewna liczba stosów. W pozycji początkowej każdy stos zawiera pewną liczbę pionków. Ruch w grze Nim polega na zdjęciu pewnej (dowolnej), niezerowej liczby pionków z dokładnie jednego stosu. Zdjęte pionki nie biorą udziału w dalszej rozgrywce. Przegrywa ten, kto nie może wykonać ruchu. Zatem w pozycji końcowej wszystkie stosy są puste, bo tylko wtedy nie można wykonać żadnego ruchu. Każda rozgrywka prowadzi do jakiejś pozycji końcowej, gdyż każdy ruch zmniejsza liczbę pionków biorących udział w grze.

W dalszej analizie będziemy korzystać z prostych własności operacji xor na liczbach całkowitych, czyli sumy modulo 2 na bitach (cyfrach w zapisie dwójkowym) tych liczb. Oto najważniejsze z nich:

- $x \operatorname{xor} 0 = x$,
- $x \operatorname{xor} x = 0$,
- $(x \operatorname{xor} y) \operatorname{xor} z = x \operatorname{xor} (y \operatorname{xor} z)$.

Ostatnia własność (łączność) pozwala nam pisać po prostu $x \operatorname{xor} y \operatorname{xor} z$.

Nim-sumą danej pozycji w grze Nim będziemy nazywać xor wielkości wszystkich stosów (wielkość stosu to liczba pionków, które się na nim znajdują). Następujące twierdzenie określa strategię wygrywającą dla gry Nim:

Twierdzenie 1 *Pozycja w grze Nim jest przegrywająca wtedy i tylko wtedy, gdy jej nim-suma jest równa* 0.

Dowód Udowodnimy następujące własności nim-sumy:

1. Nim-suma dowolnej pozycji końcowej jest równa 0.
2. Jeżeli nim-suma danej pozycji jest różna od 0, to istnieje ruch prowadzący do pozycji, której nim-suma jest równa 0.
3. Jeżeli nim-suma danej pozycji, różnej od końcowej, jest równa 0, to każdy ruch prowadzi do pozycji, której nim-suma jest różna od 0.

Punkt 1 jest oczywisty, pozostaje więc udowodnić punkty 2 i 3. Nim-sumę danej pozycji oznaczmy przez x.

Załóżmy najpierw, że $x \neq 0$. Niech i będzie numerem najbardziej znaczącego bitu x, który jest równy 1. Istnieje taki stos, że i-ty bit jego wielkości jest równy 1 — wielkość tego stosu oznaczmy przez s. Wówczas xor wielkości pozostałych stosów jest równy $x \operatorname{xor} s$. W liczbie tej i-ty bit jest równy 0, a wszystkie bardziej znaczące bity są równe tym w liczbie s. Stąd $x \operatorname{xor} s < s$. Zatem istnieje ruch, który usuwając pionki ze stosu o wielkości s, pozostawia w nim $s' = x \operatorname{xor} s$ pionków. Nim-suma po takim ruchu wynosi

$$x \operatorname{xor} s \operatorname{xor} s' = x \operatorname{xor} s \operatorname{xor} x \operatorname{xor} s = 0.$$

Załóżmy teraz, że $x = 0$. Po wykonaniu ruchu w obrębie dowolnego stosu o wielkości s, nim-suma wynosi $x \operatorname{xor} s \operatorname{xor} s' = s \operatorname{xor} s'$, gdzie s' jest nową wielkością stosu. Jeżeli $s \operatorname{xor} s' = 0$, to $s = s'$, co daje sprzeczność. Zatem każdy ruch prowadzi do pozycji, której nim-suma jest różna od 0.

Wystarczy teraz porównać punkty 1–3 z warunkami 1–3 ze strony 42, żeby się przekonać, że przegrywające są dokładnie te pozycje, których nim-suma wynosi 0. Formalnie, prosta indukcja w górę drzewa gry dowodzi tezy. ■

Gra „Staircase Nim”

Planszą do gry „Staircase Nim” jest również pewna liczba stosów, które są ponumerowane kolejnymi liczbami, począwszy od 0. W pozycji początkowej każdy stos zawiera pewną liczbę pionków. Ruch polega na przestawieniu pewnej (dowolnej) niezerowej liczby pionków z jakiegoś stosu na stos o numerze o jeden mniejszym. W końcowej pozycji wszystkie pionki znajdują się na stosie 0.

Nim-sumą pozycji w grze „Staircase Nim” będziemy nazywać xor wielkości wszystkich stosów o numerach nieparzystych. Widać tutaj wyraźną analogię do gry Nim. Każdy ruch w grze „Staircase Nim” powoduje bowiem zmianę wielkości dokładnie jednego stosu o numerze nieparzystym, a w pozycji końcowej wszystkie stosy o numerach nieparzystych są puste. Różnica jest taka, że ruch w grze „Staircase Nim” może spowodować powiększenie wielkości stosu o nieparzystym numerze.

Ale dlaczego w grze Nim nie wolno zwiększać stosów? Odpowiedź jest prosta: w przeciwnym przypadku istniałaby nieskończona rozgrywka. Jednak w grze „Staircase Nim” skończoność jest zapewniona inaczej — każdy ruch przybliża co najmniej jeden pionek do stosu 0. Zatem dla gry „Staircase Nim” jest prawdziwe twierdzenie analogiczne do twierdzenia 1:

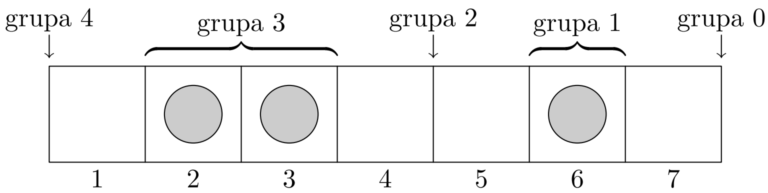
Twierdzenie 2 *Pozycja w grze „Staircase Nim” jest przegrywająca wtedy i tylko wtedy, gdy jej nim-suma jest równa* 0.

Dowód Analogiczny do dowodu twierdzenia 1. ■

Analiza gry z zadania

Zajmiemy się teraz analizą gry, której poświęcone jest zadanie.

Grupą pionków będziemy nazywać dowolny maksymalny zbiór pionków, które zajmują na planszy pola o kolejnych numerach. Maksymalność oznacza, że każda grupa pionków jest ograniczona z lewej i z prawej przez wolne pola lub końce planszy. Grupa pionków może być pusta — taka grupa znajduje się np. pomiędzy dwoma kolejnymi polami wolnymi. Grupy numerujemy od 0, począwszy od skrajnie prawej grupy.



Rysunek 1: Podział na grupy dla przykładu z treści zadania

Każdy ruch polega na przeniesieniu pewnej niezerowej liczby pionków z jakiejś grupy do grupy o numerze o jeden mniejszym. Z każdej grupy można przenieść dowolną liczbę pionków, która nie przekracza jej wielkości. W pozycji końcowej grupa 0 jest niepusta (zakładamy, że wtedy już żaden ruch nie jest możliwy). Natomiast jeżeli grupa 0 jest pusta, a grupa 1 jest niepusta, to pozycja jest wygrywająca. Wtedy każdy ruch z grupy 1 jest wygrywający, a każdy inny jest przegrywający.

Wobec tego interesować nas będą tylko takie sytuacje początkowe, w których grupy 0 i 1 są puste. Modyfikujemy drzewo gry poprzez usunięcie wszystkich ruchów prowadzących do pozycji, w których grupa 1 jest niepusta. Taka modyfikacja nie zmienia strategii wygrywającej. Natomiast otrzymujemy nową, jedyną pozycję końcową — pozycję, w której wszystkie pionki znajdują się w grupie 2.

Jeżeli w tak zmodyfikowanej grze grupy pionków potraktujemy jako stosy, a ich numery zmniejszymy o 2, otrzymamy dokładnie grę „Staircase Nim”. Zatem twierdzenie 2 określa strategię wygrywającą dla zmodyfikowanej gry, a ta, jak wiemy, nie różni się od strategii wygrywającej dla gry z treści zadania. Teraz, znając strategię wygrywającą, możemy łatwo sprawdzić, które ruchy są wygrywające, a które są przegrywające.

Rozwiązanie wzorcowe

Rozwiązanie wzorcowe dokonuje podziału na grupy i oblicza nim-sumę już podczas wczytywania danych. Pamięta tylko niepuste grupy, gdyż wszystkich grup może być zbyt wiele. Najpierw rozważane są dwa proste przypadki szczególne: jeśli grupa 1 nie jest pusta, wynikiem jest liczba pionków w grupie 1, w przeciwnym przypadku, jeśli nim-suma jest równa 0, to wynikiem jest 0.

Zajmijmy się przypadkiem, w którym grupa 1 jest pusta i nim-suma jest różna od 0 — oznaczmy ją przez x. Rozwiązanie wzorcowe sprawdza dla każdej niepustej grupy, czy istnieje z niej jakiś ruch wygrywający, czyli ruch, który powoduje wyzerowanie nim-sumy. Niech s będzie wielkością aktualnie rozważanej grupy. Jeśli grupa ma numer nieparzysty, to ruch powodujący wyzerowanie nim-sumy powinien zostawić w tej grupie $x \operatorname{xor} s$ pionków. Taki ruch jest możliwy, gdy $x \operatorname{xor} s < s$. Natomiast jeśli grupa ma numer parzysty różny od 2, to szukany ruch powinien przenieść tyle pionków do grupy o numerze o jeden mniejszym (oznaczmy jej wielkość przez t), żeby po jego wykonaniu w grupie tej znajdowało się dokładnie $x \operatorname{xor} t$ pionków. Jest to możliwe, gdy $0 < (x \operatorname{xor} t) - t \leqslant s$. W obu przypadkach z jednej grupy istnieje co najwyżej jeden ruch wygrywający. Wynikiem jest liczba tak znalezionych ruchów wygrywających.

Czas działania rozwiązania wzorcowego wynosi $\Theta(n)$. Zapotrzebowanie na pamięć jest liniowe ze względu na liczbę niepustych grup, czyli wynosi $O(n)$. Rozwiązanie wzorcowe zostało zaimplementowane w `gra.c` oraz `gra.pas`.

Testy

Poniżej znajdują się opisy testów.

- `gra1a.in` — $m = 10$, $n = 8$, przegrana, wszystkie pionki w grupie 2.
- `gra1b.in` — $m = 8$, $n = 3$.
- `gra2a.in` — $m = 8$, $n = 5$, grupa 1 zawiera 3 pionki.
- `gra2b.in` — $m = 10$, $n = 5$.
- `gra3a.in` — $m = 13$, $n = 6$, nieoczywista przegrana.
- `gra3b.in` — $m = 15$, $n = 9$.
- `gra4.in` — $m = 17$, $n = 10$.
- `gra5.in` — $m = 24$, $n = 14$.
- `gra6.in` — $m = 85$, $n = 44$, losowy.
- `gra7.in` — $m = 585$, $n = 393$, losowy.
- `gra8.in` — $m = 7166$, $n = 4027$, losowy.
- `gra9.in` — $m = 376213$, $n = 193363$, losowy.
- `gra10.in` — $m = 1906731$, $n = 801226$, losowy.
- `gra11.in` — $m = 1543149$, $n = 999183$, losowy.
- `gra12.in` — $m = 932995895$, $n = 998566$, losowy.
- `gra13.in` — $m = 940238451$, $n = 997643$, losowy.
- `gra14.in` — $m = 935204727$, $n = 999989$, losowy.
- `gra15.in` — $m = 983091535$, $n = 999240$, losowy.

Pary testów 1a i 1b, 2a i 2b oraz 3a i 3b były zgrupowane.