Paweł Mechliński, Jakub Pachocki
Treść zadania, Opracowanie

Błażej Osiński
Program

Dostępna pamięć: 64 MB. *OI, Etap III, dzień drugi, 07.04.2011*

Meteory

Bajtocka Unia Międzygwiezdna (BUM) odkryła niedawno w pobliskiej galaktyce nową, wyjątkowo interesującą planetę. Co prawda nie nadaje się ona do kolonizacji, lecz co jakiś czas nawiedzają ją deszcze nietypowych, nieznanych wcześniej meteorów.

Państwa członkowskie BUM umieściły tuż przy orbicie nowo odkrytej planety stacje badawcze, których głównym zadaniem jest zbieranie próbek przelatujących skał. Komisja BUM podzieliła orbitę na m sektorów ponumerowanych kolejno od 1 do m, przy czym sektor 1 sąsiaduje z sektorem m. W każdym sektorze znajduje się dokładnie jedna stacja badawcza, należąca do jednego z n państw członkowskich BUM.

Każde państwo wyznaczyło liczbę próbek meteorów, jakie pragnie zebrać przed zakończeniem misji. Twoim zadaniem jest, na podstawie prognozy deszczów meteorów na najbliższe lata, wyznaczyć dla każdego państwa, kiedy będzie mogło zakończyć badania.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite n oraz m ($1 \leqslant n, m \leqslant 300\,000$), oddzielone pojedynczym odstępem i oznaczające odpowiednio liczbę państw członkowskich BUM oraz liczbę sektorów, na jakie podzielona została orbita.

W drugim wierszu znajduje się m liczb całkowitych o_i ($1 \leqslant o_i \leqslant n$), pooddzielanych pojedynczymi odstępami i oznaczających państwa będące właścicielami poszczególnych sektorów.

W trzecim wierszu znajduje się n liczb całkowitych p_i ($1 \leqslant p_i \leqslant 10^9$), pooddzielanych pojedynczymi odstępami i oznaczających liczby próbek meteorów, jakich poszczególne państwa potrzebują do zakończenia badań.

W czwartym wierszu znajduje się jedna liczba całkowita k ($1 \leqslant k \leqslant 300\,000$) oznaczająca liczbę przewidywanych opadów. W kolejnych k wierszach znajdują się opisy deszczów meteorów w kolejności chronologicznej. W i-tym z tych wierszy znajdują się trzy liczby l_i, r_i, a_i (pooddzielane pojedynczymi odstępami), oznaczające, że w sektorach $l_i, l_{i+1}, \ldots, r_i$ (jeśli $l_i \leqslant r_i$) lub sektorach $l_i, l_{i+1}, \ldots, m, 1, \ldots, r_i$ (jeśli $l_i > r_i$) wystąpi deszcz meteorów, w wyniku którego wszystkie znajdujące się tam stacje badawcze uzyskają po a_i próbek skalnych ($1 \leqslant a_i \leqslant 10^9$).

W testach wartych przynajmniej 20% punktów zachodzi dodatkowy warunek $n, m, k \leqslant 1\,000$.

Wyjście

Twój program powinien wypisać na standardowe wyjście n wierszy. W i-tym wierszu powinna znaleźć się liczba całkowita w_i, oznaczająca numer deszczu, po którym stacje badawcze i-tego państwa zgromadzą łącznie co najmniej p_i próbek skalnych, lub słowo `NIE`*, jeśli dane państwo nie będzie mogło skończyć badań w przewidywalnej przyszłości.*

Przykład

Dla danych wejściowych:

```
3 5
1 3 2 1 3
10 5 7
3
4 2 4
1 3 1
3 5 2
```

poprawnym wynikiem jest:

```
3
NIE
1
```

Rozwiązanie

Wprowadzenie

Mimo dość naturalnego przedstawienia problemu i względnie nieskomplikowanego rozwiązania, korzystającego jedynie ze struktury danych, która jest już dla wielu olimpijczyków klasyczną, zadanie o deszczach meteorów było jednym z najtrudniejszych zadań finału XVIII Olimpiady Informatycznej. Wymagało ono od zawodników znalezienia zaskakującego, nietypowego zastosowania wyszukiwania binarnego, które może być także zastosowane do szerszej gamy problemów.

Rozwiązanie naiwne

Jednym z najprostszych algorytmów rozwiązujących to zadanie jest przeglądanie dla każdego deszczu wszystkich stacji, na które on spada, i aktualizowanie wymagań ich posiadaczy. Rozwiązanie takie działa w złożoności czasowej $O(mk)$, czyli zdecydowanie zbyt wolno.

Symulacja opadów

Nie ulega wątpliwości, że aby uzyskać rozwiązanie o złożoności czasowej lepszej niż $O(mk)$, będziemy musieli zrezygnować z przeglądania wszystkich stacji, na które spada dany deszcz. W tym celu możemy skorzystać z tego, że deszcze zawsze padają na stacje położone w spójnym fragmencie orbity, będzie to więc zawsze albo przedział postaci $[l_i, r_i]$, albo suma przedziałów postaci $[l_i, m] \cup [1, r_i]$. Strukturą danych, która umożliwia szybkie operacje na tego typu przedziałach, jest na przykład *drzewo przedziałowe*, opisane między innymi w opracowaniach zadań *Koleje* z IX Olimpiady Informatycznej [9] czy *Tetris 3D* z XIII Olimpiady Informatycznej [13]. Używając drzewa przedziałowego zbudowanego nad punktami $1, 2, \ldots, m$, można symulować kolejne deszcze oraz udzielać odpowiedzi na pytania o liczbę próbek zebranych dotychczas przez wskazaną stację w czasie $O(\log m)$. Niestety, ponieważ państwa mogą posiadać więcej niż jedną stację, dalej nie widać sposobu na szybkie identyfikowanie państw, które zebrały już wystarczająco wiele próbek — tutaj jedyną metodą pozostaje wciąż powolne przeglądanie wszystkich stacji.

Rozwiązanie wzorcowe

Problemy, w których należy wskazać najmniejszą wartość w spełniającą pewną własność (tutaj: najmniejsze w, dla którego po w-tym deszczu dane państwo zebrało już dostatecznie wiele próbek), często stają się prostsze, gdy zastosujemy wyszukiwanie binarne. Wtedy wystarczy znaleźć sposób udzielania odpowiedzi na prostsze pytanie: „czy dane w spełnia żądaną własność?". W naszym przypadku, dla ustalonego państwa i konkretnej wartości w nietrudno sprawdzić stosowną własność w czasie $O((m+k)\log m)$, symulując deszcze meteorów aż do w-tego i zliczając opady.

Tego typu algorytm może wydawać się niepotrzebnie skomplikowany i nieefektywny jako rozwiązanie problemu dla jednego państwa. Kluczowa różnica z opisanym wyżej algorytmem naiwnym tkwi jednak w kolejności stawiania pytań. I tu ujawnia się główny pomysł rozwiązania wzorcowego: odpowiedzi na pojedyncze zapytanie z wyszukiwania binarnego, niekoniecznie dla równych wartości w, jesteśmy w stanie udzielić w czasie $O((m+k)\log m)$ dla *wszystkich* państw naraz!

Załóżmy, że chcemy udzielić odpowiedzi na zapytania z wyszukiwania binarnego dla państw $P_1, P_2, \ldots, P_n$ i wartości wynoszących odpowiednio $w_1, w_2, \ldots, w_n$ (przy czym $w_1 \leqslant w_2 \leqslant \ldots \leqslant w_n$). Możemy wykonywać symulację z wcześniejszego podrozdziału i dla każdego i, po spadnięciu w_i-tego deszczu znaleźć odpowiedź na zapytanie dla państwa P_i — poprzez proste przejrzenie i zsumowanie liczb uzyskanych próbek ze stacji, których jest ono właścicielem. Ponieważ każdą stację przejrzymy dokładnie raz, a złożoność czasowa zasymulowania pojedynczego deszczu, jak i ustalenia liczby próbek uzyskanych dotąd przez stację, dzięki zastosowaniu drzewa przedziałowego jest rzędu $O(\log m)$, więc cała operacja zajmie w istocie czas $O((m+k)\log m)$.

Aby rozwiązać nasze zadanie, wystarczy więc przeprowadzić równoległe wyszukiwanie binarne odpowiedzi dla wszystkich państw naraz. Ponieważ etapów takiego wyszukiwania będzie $O(\log k)$, złożoność czasowa całego rozwiązania wyniesie $O((m+k)\log m \log k)$. Implementacje można znaleźć w plikach `met.cpp` i `met1.pas`.

Przykład

Przeanalizujmy na przykładzie, jak działa podana wersja wyszukiwania binarnego. Rozważmy cztery państwa, o wymaganiach odpowiednio 11, 3, 10 i 8 próbek, i pięć deszczów meteorów, zgodnie z poniższą tabelką:

piąty deszcz:			1	1	1	1		
czwarty deszcz:	4	4	4					4
trzeci deszcz:		2	2	2	2	2		
drugi deszcz:	3	3	3	3				
pierwszy deszcz:	2				2	2	2	2
numery państw	**1**	**3**	**2**	**1**	**3**	**4**	**4**	**1**

Dla każdego państwa początkowy przedział możliwych (pozytywnych) wyników to $[1,5]$. W pierwszej fazie wyszukiwania binarnego dla każdego państwa sprawdzamy wartość $w = 3$ (ciąg wartości to 3, 3, 3, 3). Za pomocą drzewa przedziałowego obliczamy, że po trzech deszczach pierwsze państwo zebrało 12, drugie 5, trzecie 9, a czwarte 6 próbek. Widzimy, że pierwsze dwa państwa są usatysfakcjonowane, a drugie dwa nie. Dlatego w drugiej fazie wyszukiwania binarnego pierwsze dwa państwa będą miały przedział wartości $[1, 3]$, a drugie dwa — przedział $[4, 5]$.

W drugiej fazie dla pierwszych dwóch państw rozważamy wartość $w = 2$, a dla dwóch pozostałych wartość $w = 4$, czyli ciąg wyszukiwanych wartości to 2, 2, 4, 4. Teraz symulujemy pierwsze dwa deszcze, po czym sprawdzamy, że pierwsze państwo zebrało 10 próbek, a drugie 3 próbki. Następnie symulujemy kolejne dwa deszcze i obliczamy, że trzecie państwo zebrało 13, a czwarte 6 próbek. Widzimy, że państwa drugie i trzecie są usatysfakcjonowane, skąd wnosimy, że przedział możliwych wyników dla nich to, odpowiednio, $[1, 2]$ i $[4, 4]$. Z kolei państwom pierwszemu i czwartemu brakuje próbek, czyli ich przedziały wartości to $[3, 3]$ i $[5, 5]$. To oznacza, że znamy już wyniki dla pierwszego i trzeciego państwa (dla czwartego jeszcze nie — musimy jeszcze sprawdzić, czy po zakończeniu wszystkich opadów będzie usatysfakcjonowane).

W ostatniej fazie wyszukiwania binarnego rozważamy ciąg wartości 1, 5 dla państw drugiego i czwartego i widzimy, że żadne z nich nie jest usatysfakcjonowane: drugie państwo nie zebrało żadnej próbki, a czwartemu brakuje jednej do żądanych 8 próbek. To oznacza, że wynikiem dla drugiego państwa jest 2, a dla czwartego odpowiedź brzmi „`NIE`". Ostateczny ciąg wyników w tym przykładzie to zatem: 3, 2, 4, `NIE`.

Testy

Nazwa	n	m	k
met1a.in	142	1 000	1 000
met1b.in	500	1 000	1 000
met1c.in	225	1 000	1 000
met2a.in	333	1 000	957
met2b.in	333	1 000	919
met2c.in	750	1 000	1 000
met3a.in	5 000	50 000	50 000
met3b.in	25 000	50 000	50 000
met3c.in	12 976	50 000	50 000
met4a.in	9 043	50 000	50 000
met4b.in	9 111	50 000	50 000
met4c.in	25 000	50 000	49 913
met4d.in	15 340	50 000	5 000
met5a.in	5 000	50 000	49 419
met5b.in	25 000	50 000	49 704

Nazwa	n	m	k
met5c.in	500	500	50 000
met5d.in	16 855	50 000	50 000
met6a.in	500	50 000	50 000
met6b.in	10 000	50 000	50 000
met6c.in	1 961	50 000	50 000
met6d.in	25 044	50 000	50 000
met7a.in	99 901	300 000	300 000
met7b.in	4	300 000	300 000
met7c.in	500	500	250 000
met7d.in	199 840	300 000	300 000
met8a.in	85 652	300 000	300 000
met8b.in	3	300 000	300 000
met8c.in	500	500	200 000
met8d.in	233 410	300 000	300 000