Krzysztof Diks, Wojciech Rytter
Treść zadania, Opracowanie

Igor Adamski
Program

Dostępna pamięć: 64 MB. *OI, etap I, 15.10–12.11.2012*

Taksówki

Bajtazar chce przejechać taksówką z miejscowości Bajtodziura do miejscowości Bajtodół, odległej od Bajtodziury o m km. W odległości d km od Bajtodziury na trasie między tymi miastami znajduje się baza taksówek dysponująca n taksówkami, ponumerowanymi od 1 do n. Taksówka numer i ma zapas benzyny wystarczający na przejechanie x_i km.

Bajtazar może się przesiadać, zmieniając taksówki. Taksówki wyruszają z bazy, ale nie muszą do niej wracać. Twoim zadaniem jest sprawdzenie, czy można przewieźć Bajtazara z Bajtodziury do Bajtodołu, a jeżeli tak, to jaka jest minimalna liczba taksówek, jakie należy wykorzystać.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite m, d oraz n ($1 \leqslant d \leqslant m \leqslant 10^{18}$, $1 \leqslant n \leqslant 500\,000$), pooddzielane pojedynczymi odstępami. Oznaczają one odpowiednio: odległość z Bajtodziury do Bajtodołu, odległość z Bajtodziury do bazy taksówek oraz liczbę taksówek znajdujących się w bazie. W drugim wierszu wejścia znajduje się n liczb całkowitych $x_1, x_2, \ldots, x_n$ ($1 \leqslant x_i \leqslant 10^{18}$), pooddzielanych pojedynczymi odstępami. Liczba x_i oznacza dystans (w km), jaki maksymalnie może przejechać taksówka numer i.

W testach wartych łącznie 40% punktów zachodzi dodatkowy warunek $n \leqslant 5000$.

Wyjście

Twój program powinien wypisać na standardowe wyjście jedną liczbę całkowitą: minimalną liczbę taksówek, którymi musi jechać Bajtazar, aby dostać się z Bajtodziury do Bajtodołu. Jeżeli nie jest to możliwe, Twój program powinien wypisać liczbę 0.

Przykład

Dla danych wejściowych:
```
42 23 6
20 25 14 27 30 7
```

poprawnym wynikiem jest:
```
4
```

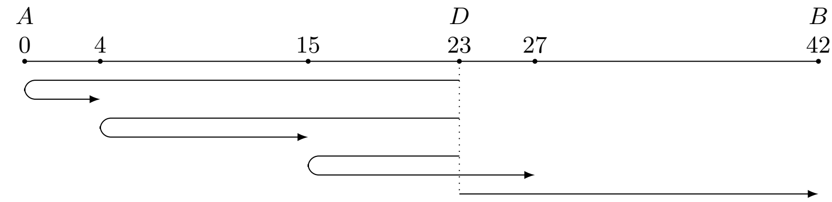
Wyjaśnienie do przykładu: *Bajtazar może jechać kolejno taksówkami numer: 4, 5, 1 i 2.*

Rozwiązanie

Celem zadania jest przewieźć Bajtazara z miejscowości A do miejscowości B. Miejscowości te są oddalone o m kilometrów. Wyobraźmy sobie oś liczbową, na której A leży w punkcie 0, a B w punkcie m.

Do naszej dyspozycji jest n taksówek, z których i-ta może przejechać x_i kilometrów; będziemy mówili, że i-ta taksówka ma *zasięg* x_i. Dla zwięzłości, będziemy też utożsamiali taksówkę z jej zasięgiem, mówiąc po prostu o taksówce x_i. Wszystkie taksówki są początkowo umiejscowione w punkcie osi o współrzędnej d, oznaczmy ten punkt przez D. Jeśli Bajtazara da się przewieźć z punktu A do punktu B, chcemy to zrobić, wykorzystując możliwie najmniejszą liczbę taksówek.

Poniższy rysunek przedstawia rozwiązanie testu przykładowego. W teście tym mamy $m = 42$, $d = 23$, a ciąg x zasięgów taksówek ma postać $(20, 25, 14, 27, 30, 7)$. W rozwiązaniu używamy kolejno taksówek $x_4 = 27$, $x_5 = 30$, $x_1 = 20$ i $x_2 = 25$.



Rys. 1: Jedno z rozwiązań testu przykładowego.

Już na podstawie analizy tego przykładu możemy poczynić pewne obserwacje. Pierwsza z wykorzystanych przez Bajtazara taksówek musi mieć zasięg co najmniej d (gdyż musi dojechać z punktu D do punktu A), a ostatnia z wykorzystanych taksówek musi mieć zasięg co najmniej $m - d$ (gdyż musi dojechać z punktu D do punktu B).

Oczywiście, taksówce niewiozącej Bajtazara na pewno nie opłaca się zawracać. Natomiast w momencie, gdy Bajtazar wsiada do jakiejś taksówki, powinna ona zacząć jechać w kierunku punktu B. Rzeczywiście, jeśli Bajtazar znajduje się wciąż przed punktem D, to w ten sposób przybliża się do każdego z punktów B i D, więc podróż w tę stronę jest jak najbardziej korzystna. Kiedy natomiast Bajtazar minie punkt D, to do dotarcia do celu wystarczy mu dokładnie jedna taksówka. Albo będzie to taksówka, w której minął punkt D, albo następna taksówka, o zasięgu nie mniejszym niż $m - d$.

Rozwiązanie wzorcowe

Odłóżmy na bok jedną taksówkę t o zasięgu co najmniej $m - d$. Jeśli żadnej takiej taksówki nie ma, od razu wiemy, że podróż Bajtazara jest niemożliwa. Natomiast jeśli jest więcej niż jedna taka taksówka, wybierzmy taką o najmniejszym zasięgu. Taksówkę t wykorzystamy jako ostatnią, będzie więc musiała dojechać do punktu B. Zatem taksówka ta może przejechać w kierunku punktu A odległość co najwyżej $\frac{1}{2}\big(t - (m - d)\big)$. Za pomocą pozostałych taksówek będziemy się więc starali dojechać do punktu o współrzędnej co najmniej $d_t = d - \frac{1}{2}\big(t - (m - d)\big) = \frac{1}{2}(m + d - t)$.

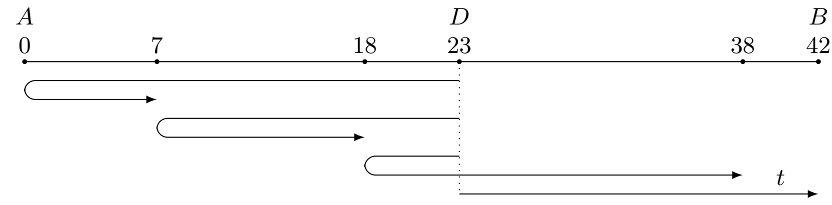
Posortujmy pozostałe taksówki *nierosnąco* względem ich zasięgów. Niech $y_1, \ldots, y_{n-1}$ oznacza tak uporządkowany ciąg zasięgów taksówek. W rozwiązaniu będziemy używać kolejnych taksówek z tego ciągu.

Niech b oznacza położenie Bajtazara przed rozważeniem i-tej taksówki. Załóżmy, że $b < m$. Jeśli $b \geqslant d_t$, to wystarczy użyć taksówki t i w ten sposób podróż Bajtazara się kończy.

W przeciwnym razie mamy $b < d_t \leqslant d$. Jeśli taksówka y_i nie jest w stanie dojechać do Bajtazara (tj. $y_i < d - b$), to stwierdzamy, że szukana trasa Bajtazara nie istnieje. Jeśli natomiast żaden z tych przypadków nie zachodzi, to używamy i-tej taksówki i przewozimy Bajtazara z punktu b do punktu $b + \big(y_i - (d - b)\big) = 2b - d + y_i$.

Zauważmy, że w ten sposób możemy też osiągnąć punkt B, nie używszy po drodze taksówki t.

Poniższy rysunek pokazuje, jak działa ta metoda dla testu przykładowego. Mamy tu $t = 20$, a ciąg y ma postać $(30, 27, 25, 14, 7)$.



Rys. 2: Schemat działania rozwiązania wzorcowego dla testu przykładowego.

Rozwiązanie wzorcowe działa w czasie $O(n \log n)$, jeśli do uporządkowania taksówek w ciąg y użyjemy efektywnego algorytmu sortowania, np. przez scalanie. Implementację można znaleźć w plikach `tak.cpp`, `tak1.pas` i `tak2.c`.

Uzasadnienie poprawności

W tej sekcji wykażemy, że jeśli rozwiązanie istnieje, nasz algorytm znajdzie rozwiązanie wykorzystujące minimalną liczbę taksówek.

Niech $z_1, \ldots, z_k$ oznacza zasięgi kolejnych taksówek w rozwiązaniu optymalnym. Jeśli jest więcej niż jedno rozwiązanie o tej samej wartości k, wybieramy ciąg taksówek największy leksykograficznie (powiemy, że ciąg $z_1, \ldots, z_k$ jest większy leksykograficznie niż ciąg $z'_1, \ldots, z'_k$, jeśli istnieje $1 \leqslant i \leqslant k$, takie że $z_1, \ldots, z_{i-1} = z'_1, \ldots, z'_{i-1}$ oraz $z_i > z'_i$). Spróbujmy rozpoznać strukturę rozwiązania $z_1, \ldots, z_k$.

Oczywiście zachodzi $z_k \geqslant t$, gdyż $z_k \geqslant m - d$, a t była taksówką o najmniejszym zasięgu nie mniejszym niż $m - d$. Ponadto możemy założyć, że $\{z_1, \ldots, z_k\}$ stanowią łącznie k taksówek o największych zasięgach. Gdyby tak nie było, istniałaby jakaś niewykorzystana taksówka x, $x > z_i$. Wówczas używając taksówki x zamiast taksówki z_i, otrzymalibyśmy poprawne rozwiązanie większe leksykograficznie niż $z_1, \ldots, z_k$.

Warto się następnie zastanowić nad kolejnością, w jakiej są ustawione taksówki $z_1, \ldots, z_k$. Poniższy lemat dostarcza kluczową własność tej kolejności.

Lemat 1. Jeśli $i < k$ i po wykorzystaniu taksówek $z_1, \ldots, z_i$ Bajtazar wciąż znajduje się przed punktem D, to $z_i \geqslant z_{i+1}$.

Dowód: Załóżmy przez sprzeczność, że przy założeniach lematu zachodzi $z_i < z_{i+1}$. Wykażemy, że wówczas ciąg taksówek $z_1, \ldots, z_{i-1}, z_{i+1}, z_i, z_{i+2}, \ldots, z_k$ również byłby poprawnym rozwiązaniem. To będzie stanowiło żądaną sprzeczność, gdyż jest to ciąg większy leksykograficznie niż $z_1, \ldots, z_k$.

Oznaczmy przez b pozycję Bajtazara po wykorzystaniu taksówek $z_1, \ldots, z_{i-1}$ i niech $\Delta = d - b$. Taksówka z_i przenosi Bajtazara do przodu o $z_i - \Delta$. Po tym ruchu Bajtazar znajduje się $\Delta - (z_i - \Delta) = 2\Delta - z_i$ jednostek od punktu D i z założenia wiemy, że nie osiągnął jeszcze punktu D. Stąd taksówka z_{i+1} przeniesie Bajtazara o $z_{i+1} - (2\Delta - z_i) = z_{i+1} + z_i - 2\Delta$ jednostek wprzód. Łącznie te dwie taksówki przeniosły Bajtazara o

$$z_{i+1} + 2z_i - 3\Delta \tag{1}$$

jednostek.

Gdyby jednak zamienić miejscami taksówki z_i i z_{i+1}, analogiczne rozumowanie pokazuje, że z ich użyciem Bajtazar albo przeniósłby się łącznie o

$$z_i + 2z_{i+1} - 3\Delta \tag{2}$$

jednostek do przodu, co jest wartością większą niż (1), albo już po wykorzystaniu lepszej taksówki z_{i+1} osiągnąłby punkt D i wtedy sama taksówka z_k wystarczyłaby mu do osiągnięcia celu. W obu przypadkach zamiana taksówek skutkuje poprawnym rozwiązaniem, co, na mocy założenia, nie może mieć miejsca. ■

Lemat 1 możemy na pewno zastosować dla dowolnego $i < k - 1$. Stąd wiemy, że $z_1 \geqslant z_2 \geqslant \ldots \geqslant z_{k-1}$. Jeśli teraz po wykorzystaniu taksówek $z_1, \ldots, z_{k-1}$ Bajtazar wciąż nie przekroczył punktu D, to na mocy lematu mamy $z_1 \geqslant z_2 \geqslant \ldots \geqslant z_{k-1} \geqslant z_k \geqslant t$. Wówczas $z_1, \ldots, z_k$ to po prostu k największych taksówek uporządkowanych nierosnąco, więc nasz algorytm znajdzie to rozwiązanie lub równoliczne rozwiązanie $z_1, \ldots, z_{k-1}, t$.

Drugi przypadek jest taki, że taksówki $z_1, \ldots, z_{k-1}$ umożliwiają Bajtazarowi przekroczenie punktu D. Wówczas wystarczy, aby $z_k \geqslant t$. Jeśli $z_k = t$, to $z_1, \ldots, z_{k-1}$ stanowią $k - 1$ taksówek o największych zasięgach poza t uporządkowanych nierosnąco, więc nasz algorytm również takie rozwiązanie znajdzie. Jeśli zaś $z_k > t$, to mamy dwa przypadki. Jeśli taksówka t występuje wśród $z_1, \ldots, z_{k-1}$, możemy zamienić tę taksówkę miejscami z z_k i w ten sposób otrzymamy poprawne rozwiązanie leksykograficznie większe (co nie jest możliwe). Jeśli zaś t nie występuje wśród $z_1, \ldots, z_{k-1}$, to musi zachodzić $z_1 \geqslant \ldots \geqslant z_{k-1} \geqslant z_k > t$, a ten przypadek rozpatrzyliśmy już wcześniej.

Rozwiązania błędne

W rozwiązaniu wzorcowym wybieraliśmy jedną taksówkę, którą oznaczaliśmy jako t, i odkładaliśmy ją na ostatnią przesiadkę. Inny pomysł na rozwiązanie mógłby zakładać, że po prostu sortujemy wszystkie taksówki nierosnąco i używamy ich w tej właśnie kolejności. Niestety jest to rozwiązanie błędne, o czym łatwo się przekonać, rozpatrując następujący przykład: $m = 25$, $d = 10$, mamy jedną taksówkę o zasięgu 15 i dowolnie wiele taksówek o zasięgu 11. Takie rozwiązanie nie uzyskiwało na zawodach żadnych punktów. Można je znaleźć w pliku `takb1.cpp`.

Innym błędem było korzystanie w rozwiązaniu tylko z liczb całkowitych 32-bitowych, które nie starczały do reprezentacji danych wejściowych. Rozwiązania wzorcowe z tym błędem uzyskiwały na zawodach 40% punktów. Takie rozwiązanie znajduje się w pliku `takb2.cpp`.