

Анализ на задача stations

Тагове: наблюдения, сегментно дърво, lazy propagation

В условието не е уточнено какво става когато няма решение, защото винаги има решение – можем да тръгнем с L литра гориво. Също ще е удобно да считаме, че накрая имаме още една станция, за която $x = L$, $f = 0$ и $r = 10^9$. Ще приемем, че в началото сортираме станциите в нарастващ ред на разстоянието x_i , така че: $x_1 \leq x_2 \leq \dots \leq x_N$ (включително с тази накрая, т.е. $x_N = L$).

Решение на първа подзадача – 7 точки

Ако $x_1 \leq F \leq r_1$, то можем да използваме първата станция за гориво и тогава ще ни стигне $L - f_1$ литра гориво, така че в този случай отговорът е $\max(x_1, L - f_1)$. В противен случай не можем да ползваме първата станция и ни трябва L литра гориво. Отговорът тук е по-малкото от двете. Това е от подзадачите, които задължително трябва да се изкарат на една задача.

Сложност: $O(1)$.

Решение на втора подзадача – 13 точки

Ограниченията ни позволяват да ползваме всички станции. Винаги е оптимално да си увеличим горивото, щом стигнем станция. Това означава, че ще използваме всички станции, стигайки до края. Можем с една симулация да намерим минималното начално гориво F , но ще обясним по-важния начин за пълното решение. Понеже се използват всички станции, то в дадена станция i преди зареждането ще имаме $F - x_i + \sum_{j=1}^{i-1} f_j$ литра гориво. Тази стойност трябва да е неотрицателна, което означава, че $F \geq x_i - \sum_{j=1}^{i-1} f_j$ за всяко i . Ако означим $\text{minf}[i] = x_i - \sum_{j=1}^{i-1} f_j$, то получаваме, че $F \geq \max_{i=1}^N \text{minf}[i]$ или отговорът е $\max_{i=1}^N \text{minf}[i]$. Ще уточним, че ако има равни x_i , то $\text{minf}[i]$ е вярно само за първото от равните, но това е без значение, защото е важно, че ще вземем предвид за първото зареждане, когато стигнем разстояние x_i .

Сложност: $O(N \log_2 N)$.

Решение на трета подзадача – 17 точки

Малкото ограничение за L ни позволява да изпробваме всички възможности стойности за F от 0 до L . Така можем да постъпим като в предната подзадача, но използвайки само станциите с $r_i \geq F$. Нека $S = \{i: r_i \geq F\}$. Достатъчно е да намерим най-малкото F , за което $\max_{i \in S} \text{minf}_S[i] \geq 0$. Ще обърнем внимание, че означението $\text{minf}_S[i] = x_i - \sum_{j \in S, j < i} f_j$.

Сложност: $O(N \log_2 N + LN)$.

Решение на четвърта подзадача – 29 точки (=17+12)

В тази подзадача N е достатъчно голямо, за да не работи предното решение, но L все още е малко, което означава, че уникалните стойности x_i също са малко. Затова можем да приложим същото решение, като разликата е, че трябва да можем бързо да намерим горивото, което ще получим от станциите на дадено разстояние x (т.е. сумата на f за тези с ограничение $r \geq F$). Това може да стане като предварително сме сортирали освен по първи приоритет x_i и по втори приоритет r_i . Така за всяко разстояние пазим кои станции сме използвали, защото с увеличаването на F , ползваме все по-малко станции.

Сложност: $O(N \log_2 N + L^2)$.

Решение на пета подзадача – 26 точки (=7+19)

Ограничението за N позволява да направим пълно изчерпване по всички възможни станции, които ще използваме. Ако сме фиксирали дадено подмножество S , то F трябва да е най-много колкото минималното r_i за $i \in S$. Освен това от по-рано знаем, че за да може да стигнем накрая с тези станции трябва $F \geq \max_{i \in S} \min f_S[i]$. Ако $\max_{i \in S} \min f_S[i] > \min_{i \in S} r_i$, то S не е валидна възможност, иначе за S минималното F ще е $\max_{i \in S} \min f_S[i]$. Отговорът накрая е минималното F , което сме открили за някое подмножество S .

Сложност: $O(N \log_2 N + 2^N N)$.

Решение на шеста подзадача – 67 точки (=7+13+17+19+11)

Интересно е, че можем да подходим подобно на трета и четвърта подзадача и да проверим много възможности за F , за да открием отговора. Интересните моменти е ясно, че са когато F е колкото някое r_i , защото това ще задава различно множество от станции, които могат да се използват. Нека за дадено F , това е множеството S . Вече знаем, че $F \geq \max_{i \in S} \min f_S[i]$, а ако това не е така значи тази възможност е невалидна, защото като повишим F вече някои от тези станции няма да бъдат използвани. Ако $F > \max_{i \in S} \min f_S[i]$, то използвайки тези станции можем да тръгнем и с по-малко начално гориво и пак да стигнем до края. При това, ако тръгнем с по-малко гориво ще може със сигурност да ползваме тези станции. Обърнете внимание, че не може да правим двоично търсене по F , защото нямаме пълна монотонност (когато горивото падне под $\max_{i \in S} \min f_S[i]$, не е ясно дали ще можем да стигнем края).

Така решението в случая е за всяко ограничение r_j да разгледаме станциите $S = \{i: r_i \geq r_j\}$. След това намираме $\max_{i \in S} \min f_S[i]$ и го сравняваме с r_j . Ако е по-голямо, то тази възможност е невалидна, иначе $\max_{i \in S} \min f_S[i]$ е най-малкото възможно начално гориво за тази възможност. Накрая отговорът е най-малката от тези стойности (ако обхождаме в нарастващ ред по r_j , то ще е директно първата възможна).

Сложност: $O(N^2)$.

Пълно решение – 100 точки

Всъщност за пълен брой точки единствено трябва да оптимизираме предната идея. Нека се пробваме да поддържаваме $\min f_S[i]$. В началото $\min f_S[i] = \min f[i]$, защото ще симулираме започването с 0 литра гориво, което пък ни позволява да ползваме всички станции. Ако разглеждаме ограниченията r_j в нарастващ ред, това, което се случва, е, че малко по-малко някои станции спират да са активни. Освен това когато станция h се премахва от множеството S на активните станции, то $\min f_S[i]$ се увеличава с f_h (за да се премахне от сметката) за $i > h$ (в сортирания ред това означава, че $x_i \geq x_h$). Всъщност тези заявки не са много трудни, ако построим сегментно дърво за максимум върху $\min f_S[i]$. Тогава заявката за промяна е просто увеличаване на всички стойности в суфикса след h , а въпросът при фиксирано r_j е просто за максимума от всички стойности в дървото (неактивните станции не ни пречат, защото можем да ги считаме като просто изискване за гориво, което трябва да се покрие за дадено разстояние). Това са стандартни заявки за сегментно дърво с *lazy propagation*.

Понеже последната техника е извън конспекта за група С, то има предвидено и алтернативно пълно решение без тази техника, възползвайки се от това, че ни интересува само максимума за цялото дърво. За да не ни се налага да променяме цял интервал, а само точки, ще разгледаме разликите между съседните стойности. По този начин вместо в дървото да имаме стойностите $\min f_S[1], \min f_S[2], \dots, \min f_S[N-1], \min f_S[N]$, ще имаме $\min f_S[1] - \min f_S[2], \min f_S[2] - \min f_S[3], \dots, \min f_S[N-1] - \min f_S[N], \min f_S[N]$. Оригиналните стойности се получават като суфиксни суми от новите. Важното е, че ако трябва да увеличим всички стойности в суфикса от i нататък, то промените на разликите са само две: $\min f_S[i-1] - \min f_S[i] - f$ и $\min f_S[N] + f$.

Така ако сегментното дърво намира максимума от суфиксните суми, имаме само точкови заявка за промяна на стойност и въпрос за глобалния максимум.

Сложност: $O(N \log_2 N)$.

Двоичното търсене е подлъгващо за тази задача, но както изяснихме по-рано нямаме пълна монотонност (ако ограничението беше долна граница, т.е. $F \geq r_i$, то тогава щяхме да имаме монотонност). За съжаление тестовете на задачата не са достатъчно силни и решения с леко модифицирано двоично търсене по F минават тестовете. Следният малък тест е контрапример:

```
3 5
1 4 1
2 0 2
3 2 3
```

Можем да се убедим, че $F = 1$ е отговорът, при $F = 2$ не можем да стигнем края, при $F = 3$ можем да стигнем края, при $F = 4$ не можем, а при $F = 5$ можем.

Автор: Илиян Йорданов (заета)