ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА
СОФИЯ, 22-24 НОЕМВРИ 2024
Група C 7-8 клас

Анализ на задача Пожарна

Първа подзадача. Фиксираме стойността на **S** и използваме пълно изчерпване, за да пробваме всяка комбинация от **K** хълма и да проверим дали те покриват всички.

Сложност – **O(N × 2^N)**.

Втора подзадача. Понеже има само една станция е достатъчно да проверим за всеки един хълм дали покрива всички и да вземем най-добрия. За проверката на хълм **i** са ни нужни най-големия и втория най-голям хълм преди и след **i**, нека техните позиции са $Lmax, Ls, Rmax, Rs$. Ако $h_{Lmax} \geq h_i$, $Lmax \neq 1$, или $h_{Ls} \geq h_i$, $Ls \neq 1$ то хълм **i** не покрива всички хълмове. Аналогично за дясната част. В противен случай хълм **i** покрива всички хълмове.

Сложност – **O(N)**.

Трета подзадача. Ако е възможно да покрием всички хълмове, то $S \leq N$. Нека фиксираме дадено **S** и намерим минималния брой хълмове, с които да покрием всички. Ако те са до **K**, то **S** е отговор на задачата.

Можем да забележим, че всеки хълм **i** покрива непрекъснат интервал от хълмове $L_i \leq i \leq R_i$. Нека $previous_i$ и $next_i$ са първите хълмове вляво и вдясно на хълм **i**, които са с височина $\geq h_i$ (ако няма такива приемаме, че са първия и последния). Аналогично нека $from_i$ и to_i са най-далечните хълмове вляво и вдясно на **i**, които са на разстояние до **S**. В случая на $t_i = 1$ те са просто **S**-тия хълм вляво и вдясно. Така имаме $L_i = \max(from_i\,, previous_i)$ и $R_i = \min(to_i\,, next_i)$.

Сега остава да изберем минимален брой хълмове. В едно оптимално покритие всеки две съседни станции ще покриват всички хълмове помежду си, или по-формално казано за станции на хълмове **i** и **j** ще е изпълнено $R_i + 1 \geq L_j$. Можем да построим граф, като свържем с ребро всяка двойка хълмове, които покриват всички помежду си. Така търсения отговор ще е минималното разстояние от хълм, покриващ хълм **1**, и хълм, покриващ хълм **N**. То може да бъде намерено с **BFS**.

Сложност – **O(N^3).**

Четвърта подзадача. Когато разстоянията са големи, не можем да циклим по **S**. Можем да забележим обаче, че ако при **S** има покритие с до **K** хълма, то и при **S+1** ще има. Също така, ако при **S** няма покритие с до **K** хълма, то и при **S-1** няма да има. Следователно можем да използваме двоично търсене по отговора, за да намерим минималното **S**.

Сложност – **O($N^2 \times \log_2(T)$).**

(*) Тук с **T** бележим разстоянието между двата крайни хълма.

Пета подзадача. Тук вече трябва бързо да намираме $previous_i$ и $next_i$. Това може да бъде направено с монотонен стек за линейно време. Сега да помислим как да решим задачата за **K=3**. Нека $prefix = \max(next_i)$ за $previous_i = 1$. Аналогично нека $suffix = \min(previous_i)$ за $next_i = N$. С други думи $prefix$ и $suffix$ са най-далечните кули, които могат да се покрият от първата и последната кула в някакво решение. Ако съществува кула **i,** такава че $previous_i \leq prefix + 1$ и $next_i \geq suffix - 1$, то имаме и кула, която да покрие всички непокрити кули в средата. В противен случай не съществува покритие.

Сложност – **O($N \times \log_2(T)$).**

ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА
СОФИЯ, 22-24 НОЕМВРИ 2024
Група C 7-8 клас

Шеста подзадача. Поради допълнително условие $previous_i$ и $next_i$ могат лесно да бъдат намерени в началото. Сложната част е в проверката за покритие с минимален брой кули. Ще представим два подхода:

I) Ще използваме грийди алгоритъм. Нека сме покрили всички кули преди **j** и кула **i** е кулата с максимална стойност на R_i, за която същевременно е в сила $L_i \leq j$. Избираме нея и продължаваме, докато $j \leq N$.

За ефективна имплементация, ще сортираме всички кули по L_i и обхождайки ги отляво надясно ще следим две неща: кулата с максимално R_i както **j**. В моментът, в който L_i стане по-голям от **j**, взимаме кулата с максимално R_i и увеличаваме отговора.

Сложност – **O(N × log₂(T)² + N × 100).**

II) Ще използваме алгоритъма от трета подзадача, като подобрим строенето на графа. Вместо да гледаме всяка двойка интервали, можем да мислим за всеки интервал като за скок от L_i до $R_i + 1$ (покриваме всички кули до R_i и отиваме в първата непокрита). Съответно всяка кула ще е ребро напред с цена 1. Понеже в някои случаи е оптимално да се „върнем“ малко назад, за да вземем интервал с голямо R_i ще построим и ребра назад $i \to i - 1$ с цени 0. В получения граф ще търсим минимално разстояние от 1 до $N + 1$. То може да се намери с **BFS** 0-1.

Сложност – **$O(N \times \log_2(T) + N \times 100)$.**

Седма подзадача. Обединяваме идеите от предните две подзадачи. За да намираме $from_i$ и to_i бързо можем да използваме двоично търсене или по-добре, показалки.

Сложност – **$O(N \times \log_2(T)^2)$.**

Осма подзадача. В грийди решението можем да заменим нормалното сортиране със count sort.

Сложност – **$O(N \times \log_2(T))$**.

Автор: Александър Гатев