Garden

Тагове	На пълното решение	На частичните решения
	Динамично оптимиране в дърво	Дейкстра

Анализ

Интересно е, че вероятно има и $O(N\ log_2\ N)$ решение със Small-to-large идеи, но би било много неприятно за измисляне и писане ☺. Също, още едно интересно нещо е, че има решение за пръчка, което е с дейкстра, но не считам за нужно да го опиша тук. Логично е да мислим, че $c_i \leq c_j$ за $(i < j)$, защото в противен случай ще може да покрием повече върхове за по-малка цена. Може да преизчислим за всеки връх минималното количество електроенергия, нужно, за да се покрият всички върхове на x разстояние от него за всяко $(0 \leq x < N)$. Когато кажа „сила x“ или подобен синоним, аз имам в предвид помпа, която работи за x минути. Когато използвам „покрит“, имам в предвид връх, който е напоен. Препоръчвам да изчетете целия анализ, макар и да сте решили задачата за някои подзадачи, защото ще спестя някои детайли, обяснени в началото.

Решения за пръчка

Задачата отвсякъде крещи за динамично оптимиране, поне се постарах тестовете да не позволяват друго решение (дано ≈ 600 реда тестов генератор да е било достатъчно). Интересното е, че и за пръчка, и за дърво, решението ще почне от една и съща базова идея, която ще се оптимизира. Също, не съм предивидил решение за $N \leq 75$, но съм готов да видя интересни резултати ☺.

Решение за $N \leq 500$

Нека за удобство да преномерираме върховете спрямо позицията им в пръчката, като най-левия да е 1, а най-десния да е N. Ще фиксираме водата която пускаме от върховете, като фиксираме първо водата на връх 1, после връх 2, …, накрая на връх N. Ще разгледаме динамично със стейт `dp[връх][баланс]`. Всички предстоящи динамични ще имат същия стейт, макар да имат различна рекурентна зависимост/оптимизации. Ако $\text{баланс} > 0$, то идва вода от предишния връх, която ще покрие всички върхове надясно до разстояние $\text{баланс} - 1$ от текущия връх, включително и самия връх. Ако $\text{баланс} \leq 0$, то предишния връх е взел вода „на заем“ от текущия връх, така че всички върхове вляво до разстояние $-\text{баланс}$ от текущия връх стоят ненапоени. Ако $\text{баланс} = 0$, то всички върхове до предишния връх ще са покрити с вода, но самия връх няма да бъде покрит. Тогава отговорът на задачата ще ни е `dp[1][0]`. Може да фиксираме колко вода ще пуснем от текущия връх и да разгледаме как се променя балансът. Разглеждането на детайлите съм го оставил за ваше упражнение ☺.

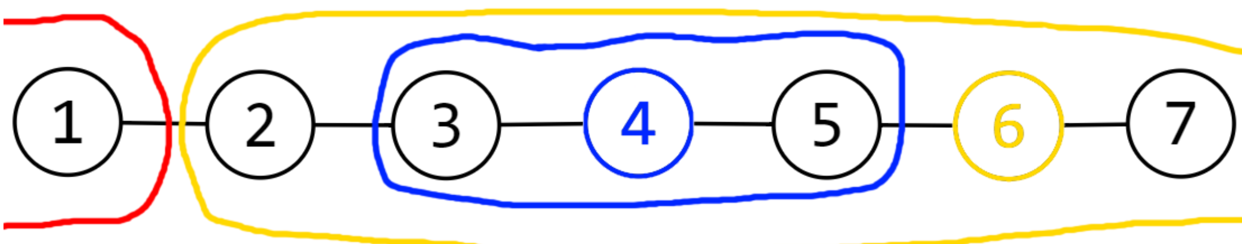
Постигната сложност: $O(N^3)$

Имплементация: `garden_23p.cpp`

Решение за $N \leq 2\ 000$

Тук може да се направят няколко оптимизации. Едни от тях са използването на сегменто дърво или дейкстра с различен стейт на динамичното, които както казах, няма да разгледам в анализа. Нека оптимизираме текущия стейт. Един вариант е да се правят забавни интервални минимуми, които честно казано, не съм разглеждал, за да мога да си спестя главоболия. Вместо това може да се приложи стандартния трик, като направим параметъра за `баланс` да е неравенство. Може лесно да се докаже, че $dp[\text{връх}][\text{баланс}] \geq dp[\text{връх}][\text{баланс} - 1]$. Заради това нека `dp[връх][баланс]` да е минималното покриване ако във текущия връх влиза вода със сила x, за която $x \leq \text{баланс}$. За целта може да разгледаме 3 варианта:

- Да покрием нужните върхове със следващия връх: $dp[\text{връх} + 1][\text{баланс} - 1]$.
- За $\text{баланс} < 0$, да покрием балансът с минималния възможен разход на електроенергия: $dp[\text{връх} + 1][-\text{баланс}] + c[\text{баланс} + 1]$, ако $\text{баланс} + 1 \leq t[\text{връх}]$.
- Да покрием върха с повече от минималния възможен разход на електроенергия: $dp[\text{връх}][\text{баланс} - 1]$. Така ще разгледаме всички случаи за пускане на вода със сила $-\text{баланс} + 2,\ -\text{баланс} + 3,\ -\text{баланс} + 4,\ \dots,\ t[\text{връх}]$. Никога не е оптимално да използваме сила $\leq -\text{баланс}$. За да докажа това, погледнете илюстрацията отдолу:



- Червено – Покритите до момента.
- Синьо – покрити от връх 4.
- Жълто – покрити от връх вдясно, в случая 6.

В случая текущия връх е 4 и $\text{баланс} = -2$. На примера е илюстриран случая, когато порkиваме със сила 2. Тъй като не покриваме всички върхове вляво с водата от връх 4, все някой връх вдясно ще трябва да ги допокрие (в случая 6). Така водата пусната от връх 4 се обезсмисля.

Така изкарваме 36 точки. Ако състезател измисли подзадачата с тази идея, той трудно би останал с 36 точки, вероятно ще стигне до 100.

Постигната сложност: $O(N^2)$

Имплементация: `garden_36p.cpp`

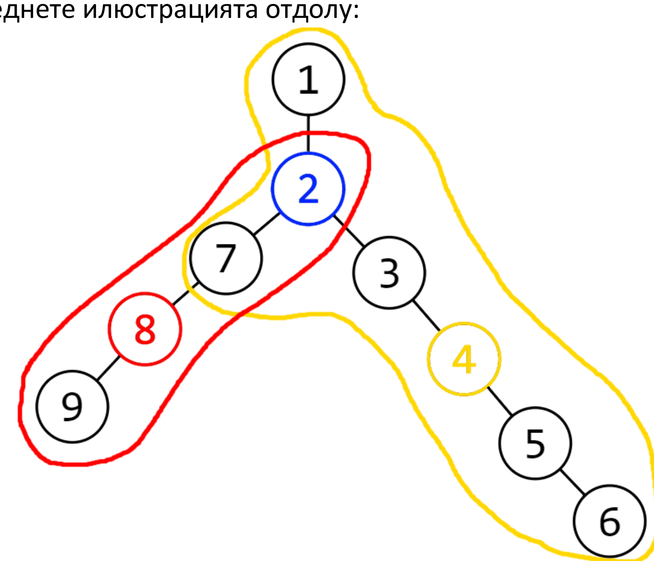
Garden

Решения за дърво

Нека навлезем в по-дълбоки води, без пояс. Дано не се нагълтате с твърде много вода ☺.

Решение за $N \leq 75$

Ще използваме динамичното dp[връх][баланс]. В дърво стейтът би значел, че родителя е отдал или взел „на заем“ вода от текущия връх. Нека да разгледаме как се променя рекурентната зависимост. Пак разглеждаме същите варианти – да пуснем или да не пуснем вода от текущия връх. Пак ще оставя по-лесните детайли за ваше упражнение ☺. Изпъква случая, където $\text{баланс} < 0$. Тогава има два случая – да отдадем взетата вода „на заем“ от текущия връх, или някой дрях връх в поддървото му. По-интересен е случая, където водата идва от поддървото. Тогава ще е нужно да излиза вода от само от децата на текущия връх. За да разгледаме случая, когато покриваме водата от ≥ 2 върха, погледнете илюстрацията отдолу:



- Синьо – текущ връх (в случая 2).
- Червено – Покритото от върха, от който излиза по-малко вода (в случая 8).
- Жълто – Покритото от върха, от който излиза повече вода (в случая 4).

Както може да се види от илюстрацията, текущия връх е 2 и $\text{баланс} = -1$. Детето от което излиза повече вода ще покрие всички върхове извън поддървото, които и ще покрие детето, от което излиза по-малко вода. Заради това няма нужда да взимаме вода „на заем“ от повече от един връх. Нищо не пречи вода да излиза от поддървото на повече от едно дете – понякога ще е нужно за покриване на самото поддърво. За целта ще фиксираме силата на водата, която ще пускаме от текущия връх/ще излиза от някое от децата му. Довършването ще е, познайте, за ваше упражнение.

Решението, вероятно по много логична причина, за която не отделих време да я осмисля, не минава тестовете за $O(N^3)$ сложност, но хваща тестовете, предвидени за $O(N^4)$ сложност.

Garden

Постигната сложност: $O(N^4)$?

Имплементация: `garden_40p.cpp`

Решение за $N \leq 500$

За подзадачата може да се напишат множество оптимизации, но не считам, че те заслужават да бъдат описани. Ако сте любопитни, може да сравните решенията ми за тази и предишната подзадача и да видите самата оптимизация.

Постигната сложност: $O(N^3)$

Имплементация: `garden_65p.cpp`

Решение за $N \leq 2\ 000$

За да оптимизираме динамичното, ще използваме същата оптимизация, която използвахме при пръчката. Случаите, които разгледаме, са идентични на решението за $N \leq 75$.

Постигната сложност: $O(N^2)$

Имплементация: `garden_100p.cpp`

П.П: По имплементационни причини, в повечето ми решения разделих дп-то на две функции, съответно `plus` и `minus`, съответно за случаите където $\text{баланс} > 0$ и $\text{баланс} \leq 0$. Параметъра на `minus` всъщност е равен на $-\text{баланс}$.

П.П №2: Задачата е стара, на относително година и половина, като претърпя множество транформации в условието, смислено и сюжетно. Ако ви е любопитно какви са, може да ми пишете и да ви покажа ☺.

Автор: Борис Михов