

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА КАМЕЛОТ

Можем да построим дърво, което моделира задачата по следния начин: окръжностите са ребра, а свързаните области между тях и вътре в тях – върхове. Тогава задачата може да се преформулира така: имаме дърво, като в някои от върховете му има хора. Хората могат да преминават от връх във връх по ребрата на дървото, като за преминаването по всяко ребро се плаща такса. Могат да бъдат отменени таксите максимум за K ребра. Искане е да се намери връх, в който могат да се съберат всички хора, като придвижването им до този връх ще бъде максимално евтино.

Тази задача има две части – геометрична и оптимизационна. Геометричната се състои в построяването на дървото и определянето на броя на хората във всеки връх.

Геометрична задача – наивно решение за $O(N^2+NM)$.

За всяка окръжност да намерим окръжността с минимален радиус, която я съдържа и да прекараме ребро от върха, съответстващ на свързаната област между двете окръжности и върха, съответстващ на свързаната област вътре в по-малката окръжност. В случай, че няма окръжност, съдържаща дадената, то ребро се прекарва между корена на дървото, който съответства на свързаната област, лежаща извън всички окръжности и върха, съответстващ на вътрешността на окръжността. За да се свърши тази работа, най-лесно е да се разглеждат окръжностите две по две и да се определи има ли ребро между тях или не. Това е алгоритъм със сложност $O(N^2)$. След това се определя в коя свързана област, колко хора, които ще ходят при цар Артур живеят. Това изисква да се срещнат всеки човек със всяка окръжност или алгоритъм със сложност $O(NM)$. Общата сложност на наивния алгоритъм за решаване на геометричната подзадача е $O(N^2+NM)$.

Геометрична задача – метод на сканиращата права – сложност $O((N+M)\log(N+M))$

За облекчаване на следващите разглеждания ще предположим, че в допълнение към всички окръжности, данните за които се четат от входа, е налице още една окръжност с център в точка $(0,0)$ и такъв радиус, че всички окръжности-крепости и точки-места на живеене на рицарите да лежат вътре в нея. В процеса на сканирането ние няма да разглеждаме влизането и излизането от тази окръжност, но дъгите от нея ще се местят из структурите от данни. По-нататък ще отъждествяваме окръжността със свързаната област, която включва всички точки, лежащи вътре в нея и не лежащи вътре в нито една окръжност, която лежи вътре в разглежданата.

Ще движим сканиращата права отляво надясно, като поместваме в двоично балансирано дърво (напр. set) горните и долните дъги (полуокръжности) на окръжностите, които пресичат правата в процеса на нейното движение. Редът в дървото (операторът за предшествоване) е по ординатата на точката на пресичане на дъгата с правата. Възможните събития и тяхната обработка са следните:

Сканиращата права се е срещнала с нова окръжност.

В този момент трябва да намерим в дървото дъгата, която е най-близо (отгоре или отдолу) до допирната точка на правата с новата окръжност. Като я намерим, проверяваме дали

окръжността, на която принадлежи тази дъга съдържа новата. Ако я съдържа, то строим ребро от нея към новата окръжност (по точно от съответните им върхове в дървото); ако не я съдържа, то „баща“ на новата окръжност ще бъде бащата на намерената най-близка окръжност. След като свършим тази работа, добавяме дъгите на новата окръжност в дървото.

Сканиращата права излиза от окръжност.

В този момент двете дъги на тази окръжност се махат от дървото.

Сканиращата права се среща с точка (място на живеене на рицар).

В този случай намираме окръжността с минимален радиус, която съдържа точката. Това става по същия начин както и при срещане на окръжност – в края на крайщата можем да разгледаме точката като окръжност с радиус 0. Разбира се, в този случай никакви дъги не се добавят и не се премахват от дървото.

Оптимизационна задача: случай $K=0$

В този случай крал Артур няма възможност да премахва такси за преминаване по ребрата и може да минимизира разходите си само за сметка на избора на мястото на съвета.

Наивно решение – $O(N^2)$

За всеки връх от дървото проверяваме какви ще бъдат разходите, ако съветът се проведе в него. За да се направи това могат да се пазят две величини за всеки връх – брой на хората в поддървото му и сумарна цена за придвижването на хората от поддървото в корена му (т.е. във върха).

Това решение се оптимизира до линейно, ако при обхождането в дълбочина се смятат посочените величини не само за поддърветата, но и за наддърветата.

Оптимизационна задача в общия случай

Нека сме фиксирали връх – евентуално място на провеждане на съвета. Да ориентираме всички ребра така, все едно, че придвижването ще бъде към него. При това положение, за всяко ребро може да се оцени колко ще икономиса крал Артур, ако направи цената на преминаване по него равна на 0. Това намаление ще бъде равно на броя на хората в съответното поддърво, умножен по цената на реброто.

По този начин можем да получим решение със сложност $O(N^2 \log N)$: Преглеждаме всички върхове като възможни места на съвета, за всеки връх правим съответното преориентиране на ребрата, чрез обхождане в дълбочина смятаме за всяко ребро, какво снижаване на цената носи нулирането на цената за преминаване по него и избираме да бъдат нулирани ония K ребра, които носят най-голямо намаление.

За да оптимизираме това решение до $O(N \log N)$ нека видим как се променя множеството от намаленията на цената по различните ребра при определяне на съседен връх за място на съвета: изменя се само намалението по реброто от старото място до новото. Така че трябва да се направи такава структура от данни, която поддържа набор от числа (може да има и еднакви) и обслужва запитвания от следните два типа:

- Замяна на на елемент от x с елемент y ;
 - Намиране на K максимални елемента.
- Може да се реализира чрез *tar*.

Оптимизационна задача в общия случай – алчен алгоритъм

Да разгледаме някое ребро (U, V) на дървото. Нека неговата цена е означена с $cost$. Да означим броя на хората в поддървото на върха U с L , а в това на върха V с R . Ще докажем, че, при оптимален избор на върха на съвета, това ребро добавя към общата сума величината $\min\{L, R\} \cdot cost$. Наистина, нека например $L < R$ и избраният връх за съвета X се намира в поддървото на U . Нека събирането на всички хора от поддървото на U в X струва A , а събирането на всички хора от поддървото на V във V , струва B . Да означим цената за преминаване на един човек по пътя от V в X с W . Тогава цената на свикването на съвета в X е $A + B + W \cdot R$, а във върха V - $A + B + W \cdot L$, което очевидно е по-малко. Така че X не е оптимален избор за върха на съвета.

Да забележим, че оценката $\min\{L, R\} \cdot cost$ не зависи от отмяната на заплащането по други ребра. Така че, достатъчно е да се пресметне тази величина за всички ребра и да се премахне таксата за K ребра най-големи стойности останалата сума е отговорът.