АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
СКАЛНО КАТЕРЕНЕ

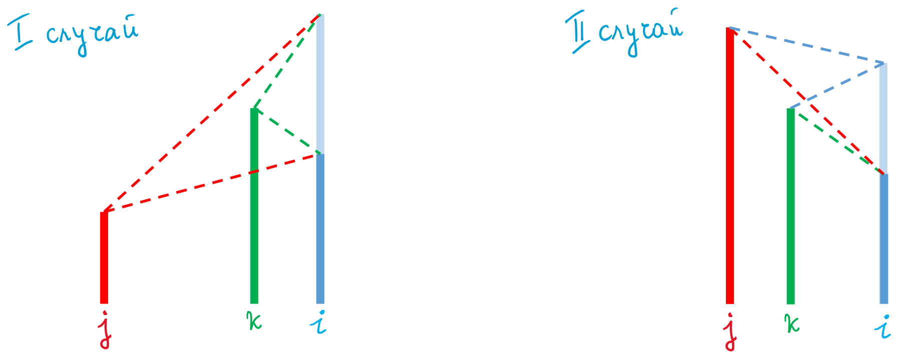
Нека първо да припомним формулата за Евклидово разстояние между две точки с координати съответно (x_1, y_1) и (x_2, y_2) – $D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Тя е единственият геометричен елемент в задачата, други знания от областта не се изискват.

Означаваме минималната дължина на въжето, което е необходимо, за да достигнем до върха на i-тата скала с R_i. Стойността на R_1 е 0, а следващите скали R_i може да изчислим, ако знаем стойностите R_j за върховете на предишните скали. Зависимостта е следната:

$$R_i = \min_j \left(\max \left(R_j, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right) \right), \text{ където } j \to i \text{ е позволен преход}$$

Простичко казано, за да пресметнем R_i, разглеждаме възможните скали j ($j < i$), от които бихме могли да опънем въже до i-тата, и избираме най-оптималната. Най-късото въже, което би свършило работа е с дължина, равна на по-голямата от R_j (за да стигнем от 1 до j) и Евклидовото разстояние между върховете на i-тата и j-тата скала. При това не трябва да забравяме, че не всички преходи от $j \to i$ са позволени, защото някои от тях може да пресичат част от друга скала.

Сега ще разгледаме случаите, в които можем да се придвижим между върховете на две скали с номера i и j.



В първия случай имаме $x_j < x_k < x_i$ и $y_j \leq y_k$ т.е. между скалите с номера j и i има друга, поне толкова висока колкото тази с номер j. Тогава никога не бихме избрали да свържем върха на скалата с номер j с този на i-тата, независимо от нейната височина. Ако скалата с номер i е по-ниска от k-тата, то такова свързване е невъзможно, защото противоречи на условието за непресичане. Ако скалата с номер i е не по-ниска от k-тата, тогава със сигурност бихме предпочели да „разбием“ връзката от j до i на две части – от j до k и от k до i, като така потенциално бихме могли да намалим дължината на необходимото въже.

Във втория случай имаме $x_j < x_k < x_i$ и $y_j > y_i$ т.е. между скалите с номера j и i има друга, по-ниска от тази с номер j. Тогава бихме предпочели да свържем върха на скалата с номер j с този на i-тата само ако тя е по-висока от k-тата. Ако скалата с номер i е по-ниска от k-тата, такова свързване или е невъзможно, защото противоречи на условието за непресичане, или е неоптимално, защото отново можем да го разбием на две по-малки части. Ако скалата с номер i е по-висока от k-тата, тогава трябва да проверим кой от двата върха е по-оптимален избор.

Оттук нататък следва да помислим как е най-удобно да поддържаме номерата на тези скали, които трябва да разглеждаме, за да определим стойностите на R_i. Трябва да можем бързо да премахваме скалите с височина до дадено h, защото, както вече видяхме, при появата на скала с такава височина, те стават безполезни. По-опитните състезатели веднага биха се насочили към идеята да пазят тези номера в стек. Във всеки един момент от изпълнението на програмата този стек трябва да съдържа редица от номера на скали, чиито върхове са подредени в намаляващ ред на техните височини.

Всички части от алгоритъма за решаването на задачата са налице, остана само да ги сглобим. Обхождаме точките от ляво надясно, при което поддържаме в стека само тези от тях, за които още не сме стигнали до друга с по-големи x- и y-координати. Нека в момента сме стигнали до върха на i-тата скала. Той трябва да извади от стека всички индекси j на точки с по-малка y-координата. Тези точки са потенциалните възможности, които трябва да разгледаме при изчислението на R_i. Съществува и още една – това е точката, която се намира на върха на стека след премахването всички с по-малки y-координати (ако не е празен). Останалите точки в стека не са оптимални, както вече разбрахме. След това върхът на i-тата скала влиза в стека и алгоритъмът продължава със следващата скала, докато не обработи и последната. Сложността е $O(\boldsymbol{N})$, защото всяка точка влиза точно по веднъж в стека.