

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА КУЛА

В основата на всички решения лежи идеята кулата да се постави на всички възможни места в редицата от сгради и за всяко място да се преброят сградите, които ще получават съобщения. Мястото, на което се поставя кулата ще определяме с израза „непосредствено след сграда с номер  $m$ “, което означава, че кулата се поставя между сгради с номера  $m$  и  $m+1$  или след сграда с номер  $N$ . Ясно е, че, ако кулата се постави преди първата сграда, то броят на сградите, които получават съобщенията от нея, ще бъде равен на 0.

### Наивно решение със сложност $O(N^2)$

Да си представим, че сме поставили кулата непосредствено след сграда с номер  $m$ . Ще броим сградите, които ще получават съобщение от нея, движейки се по редицата от кулата към сграда номер 1. Ясно е, че придвижването трябва да продължи или докато обходим всички сгради, или докато срещнем сграда, по-висока от кулата, тъй като след нея никоя сграда няма да получава сигнал. Другото нещо, което трябва да се съобрази е, че сградите, които ще получават сигнал ще бъдат разположени в наставка ред на своите височини при този начин на обхождане (от кулата към първата сграда). Това ни дава основание да броим по следния начин: поддържахме една променлива, в която пазим текущия максимум на височините на обходените сгради. Всеки път, когато този максимум се смени, т.е. срещнем сграда, по-висока от досегашната най-висока сграда, която сме срещнали, към брояча на сградите, които ще получават съобщения, добавяме 1.

Променяйки  $m$  от 1 до  $N$  и пресмятайки всеки път броя на сградите, които ще получават съобщения при позиция на кулата, непосредствено след сграда с номер  $m$ , лесно намираме максималния брой сгради, които ще получават съобщения при подходящо разположение на кулата.

Тъй като всяко обхождане на сградите от кулата до сграда 1 или до по-висока от кулата сграда е със сложност  $O(m)$  и тъй като  $m$  се променя от 1 до  $N$ , то общата сложност на алгоритъма е  $O(N^2)$ . Такова решение носи 20 точки и е реализирано във файл **tower\_n2.cpp**.

Това е най-лесния за измисляне алгоритъм, но, за съжаление, той не само, че е бавен, но не е и достатъчно „продуктивен“, т.е. не съдържа в себе си идея, която би могла да бъде развита така, че да се достигне до алгоритъм с линейна сложност (или в най-лошия случай  $O(N \cdot \log N)$ ), какъвто алгоритъм очевидно е нужен за решение за 100 точки при ограничение  $N \leq 1\,000\,000$ .

Следващото решение, което ще разгледаме също е със сложност  $O(N^2)$ , но, както ще видим, използва идея, която впоследствие ще ни доведе до линейно решение.

### Решение със сложност $O(N^2)$ , използващо стек

Отново си представяме, че сме поставили кулата непосредствено след сграда с номер  $m$ . Този път ще броим сградите, които ще получават нейните съобщения, започвайки от първата сграда и движейки последователно до сграда с номер  $m$ . Ще

казваме, че сграда с номер  $j$  покрива сграда с номер  $i$  ( $j > i$ ), ако  $h_j > h_i$  и между тях няма сграда, която е по-висока от сграда с номер  $i$ . В светлината на това определение нашата задача е да намерим максималния брой сгради, които кулата може да покрива при избор на подходяща позиция.

Идеята на това решение е, тръгвайки от началото на редицата от сгради и движейки се към кулата, постоянно да следим кои сгради вече са покрити от друга сграда – ясно е, че те няма как да бъдат покрити от кулата и да получават съобщения от нея. Когато стигнем до кулата, ще искаме да имаме списък на обходените сгради, които още не са покрити, подреден в нарастващ ред на височините им. Тогава кулата ще може да покрие ония от тях, които са с по-малка височина от нея.

За да реализираме тази идея, ще използваме стек, в който във всеки момент на обхождането ще се намират височините на онези сгради, които още не са покрити от друга сграда. Височините в стека ще бъдат подредени в нарастващ ред от върха към „дъното“ му, както ще стане ясно от следващите изречения. Нека сме стигнали до сграда с номер  $j$ . Последователно премахваме от върха на стека всички сгради (точно височините им) с височина, по-малка от тази на сграда номер  $j$  – те ще бъдат покрити от сграда номер  $j$  и не могат да получават съобщения от кулата. След като стигнем до сграда във върха на стека с по-голяма височина от тази на сграда  $j$  или до „дъното“ на стека, добавяме височината на сграда с номер  $j$  във върха на стека.

Да напомним, че сме поставили кулата непосредствено след сграда с номер  $m$ . Тогава обхождаме сградите от номер 1 до номер  $m$ , изпълнявайки описаните операции със стека. След като добавим височината на сграда номер  $m$  в стека, сме готови да преброим колко сгради ще покрие кулата, т.е. колко сгради ще получават съобщенията ѝ. За целта премахваме последователно от върха на стека всички височини, по-малки от височината на кулата (да напомним, че те са сортирани в нарастващ ред от върха към дъното), като при всяко премахане прибавяме единица към брояча на покриваните от кулата сгради.

Променяйки  $m$  от 1 до  $N$  и пресмятайки всеки път броя на сградите, които ще получават съобщения при позиция на кулата, непосредствено след сграда с номер  $m$ , лесно намираме максималния брой сгради, които ще получават съобщения при подходящо разположение на кулата.

Тъй като намирането на броя покривани сгради при всяко  $m$  е със сложност  $O(m)$  и тъй като  $m$  приема стойности от 1 до  $N$ , то общата сложност на алгоритъма е  $O(N^2)$ . Такова решение също носи 20 точки и е реализирано във файл **tower\_n2\_stack.cpp**.

Въпреки че сложността на това решение е същата, както и на предното, то е добро с това, че позволява усъвършенстване до линейно решение.

### Решение със сложност $O(N)$

За да достигнем до линейно решение, трябва по-добре да си представим как „работи“ идеята с използване на стек. Възлов тук е въпросът: „трябва ли за всяка позиция на кулата, непосредствено след сграда с номер  $m$ , когато  $m$  се променя от 1 до  $N$ , да започваме отначало пресмятането на броя сгради, които се покриват от кулата, т.е. ще получават съобщения от нея?“. Нека си представим, че имаме брояч  $Lc$  на

сградите, които се покриват от кулата. Как ще се променя стойността му, когато преместим кулата с една позиция към края на редицата от сгради и тя застане непосредствено зад сграда с номер  $m$ . Обработвайки сграда с номер  $m$ , първо вадим от стека последователно ония сгради, които са с по-малка височина от нея – те ще бъдат покрити от тази сграда и ще станат недостъпни за сигналите от кулата. Ако между тях има сгради, чиято височина е по-малка от тази на кулата, те са били преброени в някакъв момент, при влизането си в стека, като такива които кулата ще покрие, ако бъде поставена непосредствено след тях (вижте следващия параграф). Така че, за всяка сграда с по-малка височина от кулата, която се вади от стека, понеже ще бъде покрита от сграда с номер  $m$ , трябва да намалим  $Lc$  с 1.

Последното нещо, което трябва да съобразим е, че ако сградата с номер  $m$  е по-ниска от кулата, то  $Lc$  ще се увеличи с 1 в момента, в който вкарваме височината и в стека, тъй като кулата ще я покрие (между нея и сграда с номер  $m$  няма други сгради).

Тези съображения водят до алгоритъм, при който  $Lc$  се пресмята веднага след преместването на кулата с една позиция към края на редицата от сгради. Новоизчислената стойност на  $Lc$  трябва да се сравни с текущия намерен максимум на този брояч и, ако е по-голяма, да се смени текущия максимум. Когато кулата обходи всички възможни позиции, то текущият максимум ще дава отговора на задачата. Изменение в стойността на  $Lc$  може да настъпи, когато височината на сграда се вкарва или вади от стека. Тъй като височината на всяка сграда се вкарва в стека точно веднъж и се вади от стека най-много веднъж, то този алгоритъм има сложност  $O(N)$  и носи 100 точки. Такова решение е реализирано във файл **tower.cpp**.

*Автор: Руско Шиков*