**Анализ**

**Подзадача № 1**

В тази подзадача са тестовите примери. Тя е за обратна връзка от системата.

**Подзадача № 2**

За всички възможни разходки от -та до -та сграда, броим сградите в пределите им, които се виждат. Ще се виждат тези сгради, които не са закрити от двете страни с по-големи или равни по размер други сгради.

Постигната сложност:

Имплементация: sightseeing\_20p.cpp

**Подзадача № 3**

За всяка разходка от -та до -та сграда, разглеждаме всички сгради, които са в пределите ѝ. Чрез [стандартна техника със стек](https://www.geeksforgeeks.org/find-the-nearest-smaller-numbers-on-left-side-in-an-array/) намираме кои сгради от разходката не са закрити от сграда, вляво от тях, както и кои не са закрити от сграда, вдясно от тях. Нека има сграда, която се вижда и отляво, и отдясно. Тя ще е последната сграда, която не е закрита отляво, защото иначе няма да се вижда отдясно. Аналогично, тя трябва да е и първата, която не е закрита отдясно. Заради това не може да има повече от сграда, незакрита отляво и отдясно. Нека броят на видимите сгради вляво е , а на видимите отдясно е . Тогава, когато има сграда, видима и отляво и отдясно, изпитаното удоволствие ще бъде , иначе ще бъде .

Постигната сложност:

Имплементация: sightseeing\_40p.cpp

**Подзадачи №4 и № 5**

При изчислителни задачи, като тази, е удобно да разсъждаваме как всяка сграда от редицата допринася за отговора. Използвайки този подход, ние може за всяка сграда да изчислим броя разходки, при които тя се вижда. Като сумираме този брой разходки за всички сгради, ние ще получим отговора.

Броят разходки, при които -тата сграда ще се вижда, ще е равен на разликата между общия брой разходки, в които тя участва, и броя разходки, при които не се вижда.

Нека първо да изчислим **броя разходки, в които участва -тата сграда.** Може да забележим, че -тата сграда ще участва във всички разходки с начало и край . От това следва, че броят възможности за начало на разходка е , а за край на разходка е . Следователно -тата сграда ще участва в разходки.

Нека сега да изчислим и **броя разходки, при които -тата сграда не се вижда**. За целта ще означим с позицията на най-близката вляво сграда, която закрива -тата, а с – позицията на най-близката сграда вдясно, която закрива -тата. Ако няма сграда вляво, закриваща -тата, ние ще означим . Ако няма сграда вдясно, закриваща -тата, ние ще означим . Забелязваме, че -тата сграда няма да се вижда при разходки с начало и край . Затова, броят възможности за начало е , а броят възможности за край е . Следователно общият брой разходки, при които сграда с номер няма да се вижда, е равен на .

Забележете, че ако -тата сграда не е закрита от друга сграда вляво, то тя ще се вижда във всички разходки. Също, ако не е закрита вдясно, тя пак ще се вижда във всички разходки. Заради това, за двата случая ние означихме съответно и , за да може по дадената формула да се пресмята, че -тата сграда не се вижда в разходки, тоест се вижда във всички разходки.

За точки може идеята да се реализира по всякакъв начин. Зa точки е нужно да се използва [стандартната техника със стек](https://www.geeksforgeeks.org/find-the-nearest-smaller-numbers-on-left-side-in-an-array/) за намиране на и .

Постигнати сложности: и

Имплементации: sightseeing\_65p.cpp и sightseeing\_100p.cpp

*Автор: Борис Михов*