

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ОМКЛАНДИЯ (SUNSHINE)

Наивното решение е очевидно – за всяка къща p_i се търсят къщите p_l ($l < i$), p_r ($r > i$), за които ъглите склучени от векторите $\overrightarrow{p_i p_l}$, $\overrightarrow{p_i p_r}$ с вектора $(0; 1)$, сочещ нагоре, са минимални. Сборът на тези ъгли е незакритият ъгъл над къщата p_i , чрез който можем директно да изчислим слънчевите часове за къщата. Тъй като процесът на търсене на къщите p_l и p_r за всяка p_i е със сложност $O(n)$, цялостният алгоритъм е със сложност $O(n^2)$.

За да достигнем до оптималното решение на задачата трябва да се възползваме от свойството на изпъкналите многоъгълници, че ъгълът склучен между който и да е връх на многоъгълника и двата му съседни върха съдържа всички останали точки от многоъгълника. Тоест, това е най-големият ъгъл, който може да бъде склучен при този връх на полигона като връх на ъгъла. Следователно, ако имаме изпъкналата обвивка (**convex hull**) на точките $p_0, p_1, \dots, p_i, p'_0, p'_i$, където p'_0 и p'_i са съответно основите (точки от вида $(w_i; 0)$) на къщите p_0 и p_i , можем да се възползваме от това свойство. p_i и p'_i задължително са част от обвивката и ще са съседни, тъй като по условие $w_{i-1} < w_i$. Другият връх, съседен на p_i , ще бъде p_l , защото $\overrightarrow{p_i p'_i} = (0; -h_i)$, следователно ъгълът при p_i е най-големият ъгъл, който може да бъде склучен с $(0; -1)$, и най-малкият, който може да бъде склучен с $(0; 1)$ отляво. Аналогични разсъждения могат да се направят и за p_r при дадена изпъкнала обвивка на $p_i, p_{i+1}, \dots, p_n, p'_i, p'_n$. Изчисляването на изпъкналата обвивка вдясно и вляво от всяка къща p_i по отделно на всяка стъпка не е по-ефективно от наивния алгоритъм. Тук трябва да бъде направено още едно важно наблюдение – интересувани само горната дъга на изпъкналата обвивка (без точките p'_0, p'_i и p'_n), защото точките p_l, p_i и p_r нужни за изчисление на ъгъла, са в нея. Това ни позволява да използваме алгоритъма [Monotone Chain](#), за да изчислим горната дъга и всички ъгли с $O(n)$ сложност (точките са сортирани по w координатата си). Тъй като алгоритъмът е итеративен, на всяка стъпка i (добавяне на къщата p_i към дъгата) получаваме горната дъга на първите i къщи, откъдето се получават p_l / p_r и съответно по-малкото от лявата / дясната част от ъгъла и 90° . Важно е да се отбележи, че този алгоритъм трябва да бъде пуснат два пъти – веднъж от p_0 към p_n за лявата част от ъгъла, и втори път от p_n към p_0 за дясната. След това изчисляването на целия ъгъл и часовете е тривиално. Цялостна сложност: $O(n)$.

Автор: Стефан Петров