АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ОМКЛАНДИЯ (SUNSHINE)

Наивното решение е очевидно – за всяка къща $\boldsymbol{p_i}$ се търсят къщите $\boldsymbol{p_l}$ $(\boldsymbol{l < i}), \boldsymbol{p_r}$ $(\boldsymbol{r > i})$, за които ъглите сключени от векторите $\overrightarrow{\boldsymbol{p_i p_l}}, \overrightarrow{\boldsymbol{p_i p_r}}$ с вектора $(\boldsymbol{0}; \boldsymbol{1})$, сочещ нагоре**,** са минимални. Сборът на тези ъгли е незакритият ъгъл над къщата $\boldsymbol{p_i}$, чрез който можем директно да изчислим слънчевите часове за къщатата. Тъй като процесът на търсене на къщите $\boldsymbol{p_l}$ и $\boldsymbol{p_r}$ за всяка $\boldsymbol{p_i}$ е със сложност $\boldsymbol{O(n)}$, цялостният алгоритъм е със сложност $\boldsymbol{O(n^2)}$.
За да достигнем до оптималото решение на задачата трябва да се възползваме от свойството на изпъкналите многоъглъници, че ъгълът сключен 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 сключен при този връх на полигона като връх на ъгъла. Следователно, ако имаме изпъкналата обвивка (**convex hull**) на точките $\boldsymbol{p_0, p_1, \ldots p_i, p'_0, p'_i}$ **,** където $\boldsymbol{p'_0}$ и $\boldsymbol{p'_i}$ са съответно основите (точки от вида $(\boldsymbol{w_i}; \boldsymbol{0})$) на къщите $\boldsymbol{p_0}$ и $\boldsymbol{p_i}$, можем да се възползваме от това свойство. $\boldsymbol{p_i}$ и $\boldsymbol{p'_i}$ задължително са част от обвивката и ще са съседни, тъй като по условие $\boldsymbol{w_{i-1} < w_i}$**.** Другият връх, съседен на $\boldsymbol{p_i}$**,** ще бъде $\boldsymbol{p_l}$**,** защото $\overrightarrow{\boldsymbol{p_i\, p'_i}} = (\boldsymbol{0}; \boldsymbol{-h_i})$**,** следователно ъгълът при $\boldsymbol{p_i}$ е най-големият ъгъл, който може да бъде сключен с $(\boldsymbol{0}; \boldsymbol{-1})$, и най-малкият, който може да бъде сключен с $(\boldsymbol{0}; \boldsymbol{1})$ отляво. Аналогични разсъжения могат да се направят и за $\boldsymbol{p_r}$ при дадена изпъкнала обвивка на $\boldsymbol{p_i, p_{i+1}, \ldots p_n, p'_i, p'_n}$**.** Изчисляването на изпъкналата обвивка вдясно и вляво от всяка къща $\boldsymbol{p_i}$ 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 $\boldsymbol{p'_0}$, $\boldsymbol{p'_i}$ и $\boldsymbol{p'_n}$)**,** защото точките $\boldsymbol{p_i}$ **,** $\boldsymbol{p_l}$ и $\boldsymbol{p_r}$ 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 $\boldsymbol{O(n)}$ сложност (точките са сортирани по **w** координатата си). Тъй като алгоритъмът е итеративен, на всяка стъпка **i** (добавяне на къщата $\boldsymbol{p_i}$ към дъгата) получаваме горната дъга на първите **i** къщи, откъдето се получават $\boldsymbol{p_l}$ **/** $\boldsymbol{p_r}$ 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 $\boldsymbol{p_0}$ към $\boldsymbol{p_n}$ за лявата част от ъгъла, и втори път от $\boldsymbol{p_n}$ към $\boldsymbol{p_0}$ за дясната. След това изчисляването на целия ъгъл и часовете е тривиално. Цялостна сложност: $\boldsymbol{O(n)}$**.**

Автор: Стефан Петров