

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ИЗПЪКНАЛА ОБВИВКА

Първото нещо, което можем да съобразим е, че всички операции по премахване на няколко точки могат да бъдат разбити на елементарни операции по премахването на една точка. Тъй като всяка операция по добавяне добавя точно една точка и операциите са общо N , то не може да има повече от N елементарни операции по премахване на една точка

Нека вече сме намерили изпъкналата обвивка на множеството от точки в определен момент. Да разгледаме отсечката, която свързва най-лявата и най-дясната точка от множеството. Частта от изпъкналата обвивка, която лежи над тази отсечка заедно с двата края на отсечката да наречем горна част на изпъкналата обвивка. Другата част от изпъкналата обвивка заедно с двата края на отсечката да наречем долна част на изпъкналата обвивка. По този начин двата края на отсечката лежат принадлежат едновременно на горната и долната част на изпъкналата обвивка. Горната и долната част заедно отсечката на изпъкналата обвивка се явяват изпъкнали многоъгълници. Лицето на цялата изпъкнала обвивка е равна на сумата от лицата на тези два многоъгълника. Така че е достатъчно да се научим да изпълняваме всички операции с горната част от изпъкналата обвивка – за долната ще бъде аналогично.

Решение за 50 точки

Ще поддържаме стек, в който ще се поместват точките от горната част. Ще добавяме нова точка в стека по същата логика както в алгоритъма на Грахам. А именно, премахваме точките от върха на стека докато завоят между отсечката от двете точки във върха на стека (да ги наречем A и B) и новата точка (D) не стане десен (т.е. скаларното произведение $AB \times AD$ не стане <0). След това добавяме новата точка в стека. Заедно с построяването на горната част на изпъкналата обвивка можем да намираме и лицето ѝ като заедно със всяка точка, точка, която се намира в стека да пазим лицето на изпъкналия многоъгълник, който се образува от нея и всички точки, които се намират под нея в стека. За да се получи това е достатъчно при добавянето на нова точка във върха на стека, към лицето, което се съхранява с текущата точка във върха на стека да се добави лицето на триъгълника, образуван от двете точки във върха на стека и новата точка.

Вече можем да добавяме точка към горната част на изпъкналата обвивка с едновременно изчисляване на лицето ѝ. Описаният по-горе алгоритъм не позволява премахване на точка. Добавянето на нова точка в стека се предшества от премахването на няколко точки от върха на стека. При премахването на тази точка тези точки трябва да бъдат върнати в стека. За целта ще използваме още един стек, в който ще вкарваме точките, които вадим от първия. Когато трябва да върнем няколко точки в първия стек, то ще ги вземаме от втория. За целта трябва при всяко добавяне на нова точка в първия стек допълнително да се запазва размера на другия стек, за да се знае колко точки трябва да бъдат върнати от втория в първия стек, когато тази точка се премахва.

С такъв алгоритъм могат да се обработват всички видове операции. Той има сложност $O(N \cdot M)$, където M е броят на редовете от входа, в които се изисква премахването на няколко точки от върха на стека. Това решение ще получи 50 точки.

Решение за 100 точки

За да направим по-бързо решение, трябва да си представим коя е най-трудоемката част от предния алгоритъм. Да разгледаме следната ситуация: нека е конструирана горната част на изпъкналата обвивка като множество от точки (намират се в стека). Нека добавяме нова точка, която лежи по-високо (и по-надясно, тъй като

абсцисата ѝ е по-голяма по условие) от всички други точки. Тогава трябва да премахнем от стека всички точки, освен първата. Сега нека премахваме тази точка – тогава трябва всички точки отново да се върнат от втория стек в първия. По този начин, за тези две операции алгоритъмът извършва $2*N$ действия. Тъй като това може да се повтаря много пъти, то сложността ще бъде $O(N^2)$.

Да заменим в нашето решение стека с обикновен масив, за който допълнително ще пазим броя точки, които се намират в горната частна изпъкналата обвивка (размера на стека). При добавянето на нова точка ще намираме с двоично търсене броя точки, които трябва да премахнем от стека преди в него да добавим новата точка. Вместо да местим всички тези точки в другия стек, ние ще запишем в него само размера на горната част, а също така и информация за крайната точка от множеството точки, които се премахват (номерът ѝ в масива, координатите ѝ, а също и лицето на многоъгълника, който е свързан с нея). След това ние ще заменим тази точка с новата, изчислявайки за нея лицето и размера на горната част. При това всички други точки, които са били премахнати от горната част, остават в масива без промени (просто те се намират извън границите на горната част). При премахване на точката от върха на стека (в стария алгоритъм), е достатъчно само да се измени размера на горната част и да се замени точката с тази, която пазим във върха на втория стек. Сложността на такъв алгоритъм е $O(N\log N)$.

Автор: Руско Шиков