

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА КОЛИ

Автор: Антон Анастасов (aanastasov@mit.edu)

Най-тривиалното решение на задачата изисква изпробването на всеки от свободните $L - N$ сектора и решаването на подзадачата, в която може да намерим колко време би отнело, преди колата на Петър да бъде задмината, ако започне от някоя фиксирана позиция с номер $1 \leq p \leq L$. За решаването на тази подзадача съществуват множество алгоритми:

- Ако забележим, че относителният ред на колите, които участват в състезанието, не се променя и единствено някои коли биват дисквалифицирани, като останалите запазват релативния ред помежду си, може да симулираме процеса на изпреварване с алгоритъм със сложност $O(N^2)$: достатъчно е да разгледаме всяка двойка от съседни коли и да сметнем първия момент, в който предната кола ще бъде изпреварена от задната кола (което е възможно единствено когато задната кола е с по-висока скорост). Намирането на първото изпреварване отнема $O(N)$, ако имаме колите, сортирани спрямо начална позиция. След това може да премахнем изпреварената кола и да продължим по аналогичен начин до момента, в който изпреварената кола е тази на Петър.

- Може да забележим, че редът, в който се случват изпреварванията, е без значение и може да сметнем за всяка от останалите N необходимото време да изпревари колата на Петър. Най-малкото такова време съответства на времето, за което Петър може да участва преди да бъде дисквалифициран. Тази идея съответства на алгоритъм със сложност $O(N)$ за тази подзадача.

Горните два алгоритъма за дадена подзадача, комбинирани с факта, че има не повече от L позиции, които да бъдат изпробвани, води до алгоритми със сложност $O(LN)$ и $O(LN^2)$, които водят до решения за 20 и 10 точки съответно.

Ако забележим, че винаги е оптимално да поставим колата на Петър непосредствено зад една от другите съществуващи коли, то може да намалим броя на позиции, които разглеждаме до N , и съответно горните две решения стават $O(N^2)$, даващо 40 точки, и $O(N^3)$, даващо 20 точки. Трябва да се отбележи, че има един частен случай, в който има повече от N позиции, които трябва да бъдат разгледани – случаят, при който колата на Петър е измежду най-бързите коли; тогава няма значение къде ще бъде поставена колата и всяка позиция е оптимална – в този случай е достатъчно да изведен най-малките незаети позиции (първите 100, ако са повече от 100).

За да решим задачата изцяло е нужен алгоритъм със сложност $O(N \log N)$. Един такъв алгоритъм използва повечето от идеите на бавните алгоритми. Преди да опишем цялостен алгоритъм за решаването на задачата, нека разгледаме идеята за симулация и нека се опитаме да оптимизираме алгоритъма за симулация на една начална позиция до такъв със сложност $O(N \log N)$. Вместо да намираме първия момент, в който ще стане изпреварване с линейно обхождане на всяка двойка от съседни коли, може да построим двоична пирамида (binary heap) от тези събития. За да бъдат всички

изчисления точни, вместо да използваме floating point типове, използваме дроби (като не е нужно те да бъдат съкращавани, понеже единствено ще ни е нужно да сравняваме дали една дроб е по-малка от друга). Нека приемем, че на върха на пирамидата е събитието, отговарящо на първото изпреварване. След премахването на този елемент е нужно да обновим пирамидата така, че тя да съдържа всички възможни изпреварвания на останалите коли (които са с една по-малко заради първата *изпреварена* кола, която съответства на събитието, което сме премахнали от върха на пирамидата). Единственото събитие, което потенциално не е в пирамидата, е изпреварването на колата пред *изпреварената* кола от тази, която е зад *изпреварената* кола – т.е. новата промяна е локална. За това е достатъчно просто да проверим дали предната кола е по-бавна от задната и ако това е така, да добавим в пирамидата момента на това изпреварване. За да можем бързо да намираме кои са колите пред и зад всяка кола може да построим двойно-свързан списък, който да обновяваме след всяко изпреварване, като премахваме един елемент от списъка (този, който съответства на *изпреварената* кола) за константно време. След тези три промени (премахване на елемента на върха на пирамидата, добавяне на един нов елемент, и премахване на един елемент от двойно-свързания списък) всичко е идентично с началното състояние на структурите и може да продължим по аналогичен начин.

Трябва да се отбележи, че в така описания алгоритъм ние се интересуваме от момента, в който колата на Петър бива изпреварена и, понеже пирамидата е по времената на тези събития, това е лесно.

За цялото решение е достатъчно да забележим, че може да поставим в пирамидата N копия на колата на Петър – по едно зад всяка от колите на другите състезатели. Понеже всички тези копия ще имат една и съща скорост, те не могат да се изпреварват и съответно не променят момента, в който ще бъдат изпреварени. Тази идея води до алгоритъм със сложност $O(N \log N)$ за цялата задача. Подзадачите са направени така, че решение със сложност $O(N\sqrt{N})$ да получи 60 точки, а решение със сложност $O(N \log^2 N)$ да получи 80 точки.