

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА КАРТИ

Тази задача първоначално беше предвидена за Div1 900/950 в TopCoder. Решението е базирано на доста стандартна техника, но приложена по нестандартен начин. Хващането на 100 точки е предвидено само за наистина най-добрите (стига да се дадат Time Limit-ите и Memory Limit-ите, за които помолих), но постигане на 70-80 точки би трябвало да не затрудни добрите състезатели в A група.

Първо, ако не знаем какво да правим, то правим брутфорс. Около 20-25 точки могат да се хванат с прост брутфорс, който пробва всички възможности.

По-опитните състезатели трябва веднага да са съзряли динамичното оптимизиране в задачата (оптимален резултат, има лесен брутфорс / рекурентна зависимост, която би решила задачата, макар и бавно). Въпросът е как точно да го направим.

Ако ползваме STL-ски мап, веднага точките ни скачат на 30-40. Ползването на `hashmap` пък би ни дало около 50-60 точки. Но и двете са един вид "чийт" за хората, които не могат да измислят истинският стейт, който може да се пази в обикновен масив.

Все пак, как да достигнем до заветните 100 точки?

Най-очевидният стейт е шестмерно динамично (47)(52)(52)(52)(52)(52) – до коя карта сме стигнали и кои са петте карти в нас. След малко мислене можем да забележим, че е достатъчно да имаме само (47)(52)(52)(52)(52) (до коя карта сме стигнали и 4 от картите) или (52)(52)(52)(52)(52). Тъй като първо изхвърляме стара карта и после взимаме новата, то последно взетата карта със сигурност е в нас. Така по картите в ръката ни можем да видим докъде сме стигнали в купчинката. Това обаче все още изисква твърде много памет.

С много мислене можем да забележим, че с 4 карти можем кодираме 5-тата. Това всъщност е интересна логическа задача, чието решение можете да видите тук: <http://www.espr1t.net/prog101/problems.html#Problem58>. Така постигаме динамично (52)(52)(52)(52), което е и достатъчно малко за да решим задачата.

Това решение, обаче, макар и подходящо за човек, е сравнително сложно (по-скоро бавно) за изпълнение на компютър. Вместо това ще ползваме динамично с по-нестандартно кодиране на стейта.

Вече казахме, че единственото, което ни интересува, са петте карти в текущата ни ръка. Колко стейта имаме? 5 карти, от 52 възможни са $52! / (5! * 47!) = 2,598,960$. Под три милиона стейта! Тъй като на всеки стейт можем да изхвърлим само една от 5-те карти, то броят преходи от всеки стейт е 5. Така общият брой операции се очаква да е около 15 милиона, ако можем да направим всеки преход в една операция.

Но въпросът как да кодираме тези стейтове остава. Всъщност, дали? Нали токущо сметнахме с комбинаторика колко стейта има? Защо не ползваме комбинаторика за да кодираме и самите стейтове? За всеки 5 карти в ръката ни съществува точно един от тези стейтове. Не можем ли бързо да намерим кой стейт отговаря на тези 5 карти? Всъщност се оказва, че можем. Така просто подаваме на рекурсията 5-те карти, с които сме в момента (което ни показва и колко карти вече сме изтеглили от купчинката), кодираме ги в едно число между 0 и 2598960 с комбинаторика, и индексираме с това число в динамичната таблица.

Разбира се, за да постигнем това, трябва да докараме до що-годе константни преходи. Не можем с една единствена операция, но все пак можем да постигнем $O(1)$ за прехода. За целта трябва:

1. да преизчислим колко точки бихме спечелили, имайки 5 карти;
2. да преизчислим $S(N, K)$ за да кодираме бързо стойта;
3. да пазим картите в удобен за горните две неща начин.

Преизчислението на 2 е тривиално. Преизчислението на 1 е сравнително лесно, ако измислим хубаво 3.

Всяка карта е число между 0 и 51, включително. Сравнително удобно за компютрите е това, че има 4 цвята. Така най-младшите 2 бита могат да отговарят за цвета, а следващите 4 бита – за силата на картата. Ако имаме карта `card`, за да вземем нейния цвят правим `(card & 3)`, а за да вземем нейната сила правим `(card >> 2)`.

Преизчислението на 1 можем да направим с два масива:

1. `ws(4)(4)(4)(4)(4)`, указващ колко бихме спечелили ако боите на петте карти в ръката ни са някакви_пет_числа_между_0_и_3.
2. `wr(13)(13)(13)(13)(13)`, указващ колко бихме спечелили ако силата на петте карти в ръката ни са някакви_пет_числа_между_0_и_12.

Така решението ни прави около $((N+5)! / (5! * N!)) * 10$ операции, което, изразено като сложност, е $O((N+5)! / N!)$.

Тестовите на задачата са така съставени, че градивно да увеличават броя карти в купчинката за да се разграничат различни решения. А различни решения наистина има, аз успях да измисля 9 такива.

Автор: Александър Георгиев