

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА РОЖДЕНИ ДНИ

Първо, повече информация за Birthday Paradox можете да намерите тук: [https://en.wikipedia.org/wiki/Birthday\\_problem](https://en.wikipedia.org/wiki/Birthday_problem)

А сега за нашата задача (както можете да видите, тя е надграждане над оригиналния проблем). Може би по-стандартна задача би било да питаме “Ако има  $K$  човека, какъв е шансът поне  $N$  от тях да имат рожден ден на един и същи ден?”. Можем да забележим, обаче, че ако можем да си отговорим на горния въпрос (тоест можем да го решим), то може да решим и така поставената задача – просто като пробваме различни  $K$ .

Привиден проблем е, че не знаем колко голямо може да стане  $K$ . Все пак, от примера в условието изглежда, че няма да е толкова много (ако за 13 е под 2000, бихме очаквали под 20 да е също относително малко). Тук исках от състезателите да си напишат програма (потенциално бавна и/или гълтаща много памет), която да изпълняват на машините си за да намерят, че най-големият отговор е под 3500.

Всъщност, как решаваме “Ако има  $K$  човека, какъв е шансът поне  $N$  от тях да имат рожден ден на един и същ ден?”. Реално, по-лесният за решаване въпрос е “какъв е шансът да НЯМА ден с поне  $N$  рожденика”, като от неговия отговор лесно намираме оригиналния (просто изваждаме намерения шанс от 1).

По-опитните състезатели веднага ще надушат динамичното оптимизиране тук. Динамичното е със стейт [до кой ден все още не сме имали  $N$  рожденика][колко човека все още не са имали рожден ден]. Вътре за конкретния ден, който разглеждаме, трябва да решим колко хора ще имат рожден ден, което правим с цикъл от 0 до  $N - 1$ .

Всичко би било много просто (твърде просто, за  $A$  група всъщност) ако нямаше лек проблем – как намираме какъв е шансът в конкретен ден в годината да има *точно*  $X$  рожденика (което въртяхме в цикъла от 0 до  $N - 1$ )? Единият вариант е с комбинаторика (което обаче не е хубав вариант, тъй като много лесно може да изгубим точност, работейки с много големи числа за междинните сметки). Другият (и предпочитан от мен) вариант е да направим второ динамично, което смята това. То пък може да има стейт [кой ден в годината разглеждаме][колко рожденика искаме да имаме][колко човека все още не са имали рожден ден].

Така първото динамично има стейт  $[365][K]$ , като вътрешно има цикъл до  $N$ . Неговата сложност е  $O(365 * K * N)$  – макар и на теория да е “грешно” да пишем константа в сложността, тъй като в случая тя е доста голяма, сме я показали.

Второто динамично има стейт  $[365][N][K]$ , като вътре в тялото си прави само константни операции. Тоест неговата сложност също е  $O(365 * K * N)$ . Двете динамични са независими едно от друго (едното ползва резултатите от другото, но можем да изпълним първо едното и после другото), съответно сложността за цялото решение е също толкова.

Последно, за да не се притеснявате от мемори лимит или тайм лимит можеше на работната ви машина да преизчислите всички отговори (в крайна сметка в задачата има точно 20 възможни входа!) и просто да печатате съответното число, постигайки сложност  $O(1)$ .

Автор: Александър Георгиев