

Achievement Unlocked

(Решение)

Първата по-сложна задача в темата беше Achievements (или Achievement Unlocked). Тя беше подвеждаща по няколко причини, но също така частично стандартна, така че не би трябвало да е твърде костелив орех за най-добрите състезатели.

Искат ни някаква оптималност. И има цели числа, които не са твърде много на брой. Добрите състезатели трябва веднага да са надушили динамично оптимизиране, което всъщност наистина се изисква в задачата.

Стейтът, обаче, не е чак толкова очевиден. Тъй като не можем просто да пазим информацията до кой ачийвмънт е стигнато във всяко от нивата, както и да пазим цялото време, което има Ели, трябва да сме малко по-хитри.

Първото наблюдение, което трябва да направим, е да видим, че 125,000 минути са ни излишно много. Въпреки че $50 \text{ (нива)} * 50 \text{ (ачийвмънта)} * 50 \text{ (минути за игра)}$ дава точно толкова, то реално няма да ползваме повече от $50 \text{ (ачийвмънта)} * 50 \text{ (минути за игра)}$. Тъй като 50 минути на игра е максимум, то ще можем да изиграем всяко от нивата с ачийвмънт за толкова време. Започвайки от най-сложните нива ще можем да вземем ачийвмънтите и за по-простите такива безплатно, като така се оказва, че ако Ели има 2500 минути или повече, то тя може (гарантирано) да вземе всичко.

Втората хитрост е, че можем да приложим малко по-умна стратегия от това да играем нивата в произволен ред.

Донякъде очевидно е, че спокойно можем да играем нивата в нарастващ ред на сложност. Така гарантирано ще можем да изиграем всяко от тях по максимален брой пъти. Като цяло яко. Това, обаче, не ни помага да избегнем проблема с големия стейт, тъй като трябва да пазим кои от тях са изиграни и по колко пъти, за да можем да изчисляваме отговора правилно.

Да видим какво става, ако играем нивата в намаляващ ред. Изглежда, че е окей, тъй като можем да пазим следния стейт: на кое ниво сме в момента, колко пъти сме изиграли нива с по-голяма сложност и колко време ни остава. Така стейтът може да бъде $[50][51][2500]$, което е сравнително малко откъм памет, а също така дава еднозначен отговор оттук нататък. Всъщност той е и достатъчен за да решим задачата.

Тук идва и много подвеждащото нещо. Макар и стейтът да е окей, той ни навява на мисълта, че ще играем нивата в намаляващ ред на сложност. А това не винаги е оптимално. Следният малък тест показва това:

3	15				
100	3	100	100	100	
5	1	1			
5	2	1	1		

Ако изиграем два пъти най-сложното ниво, то вече второто ниво ще е останало без отключен ачийвмънт и няма да можем да го изиграем, въпреки, че имаме времето за това. Така намереният от нас отговор би бил 203. Ако изиграем първо второто ниво, после два пъти третото, обаче, ще получим отговор 303.

Без да променяме стойта, ще позволяваме да играем дадено ниво (до броя ачийвмънти в него) дори ако "вече" сме изиграли някакви по-сложни нива. Това е еквивалентно на първоначалната ни идея да играем нивата в нарастваща сложност, като, обаче, правим динамичното в изглеждащо намаляваща. Малко втф момент, което е и сложното (готиното?) нещо в задачата =)

Вътре във всеки стойт трябва да решим колко пъти ще изиграем текущото ниво. Това може да е между 0 и броя ачийвмънти на нивото, включително. Хмм, така, обаче, се появява нов проблем. Какво става, ако до сега сме изиграли 42 нива и решим да изиграем текущото 13 пъти? Така общият брой изиграни нива става 55, тоест > 50 , което ни прецаква стойта. С малко мислене виждаме, че това всъщност не е от значение - за по-лесните нива няма значение дали сме изиграли 50 или 55 от по-сложните. Можем да ограничим втория аргумент на динамичното до 50 без да загубим информация.

В тялото на динамичното правим $O(Q_i)$ операции, където Q_i беше броят ачийвмънти на нивото. Тъй като и то е ограничено до 50, то цялата сложност на алгоритъма ни е $O(N * Q_i * \min(M, 2500))$ (за таблицата) * $O(Q_i)$ (за тялото), или $O(N * Q_i^2 * \min(M, 2500))$.