Spaceship
(решение)

Задачата *Spaceship* изискваше само стандартни неща като брутфорс, симулация и свързани списъци. Само че както брутфорсът, така и свързаните списъци не бяха особено очевидни.

Първото нещо, което трябва да направим преди да започнем с каквото и да било решение, е да си напишем симулация – като изберем даден кораб за начален, как се развива играта (за да намерим колко секунди ще продължи тя). Това е тривиално и не би трябвало да затрудни участниците в тази група.

По-опитните състезатели интуитивно биха тръгнали да търсят нещо като динамично или БФС. Само че в този случай интуицията лъже – тъй като дъската се променя, динамичното няма как да работи. Алчните стратегии също не биха свършили работа. Пълното изчерпване не изглежда вероятно, тъй като нямаме рекурсия (от всеки кораб е еднозначно определено какво правим). Какво, тогава правим?

Оказва се, че тук може да приложим малко по-нестандартен брутфорс. Вместо да правим рекурсия (както би трябвало да сме свикнали като чуем "брутфорс"), просто ще разгледаме *всяка* възможна начална позиция (до N * M на брой) и ще пуснем *пълна симулация*. Тъй като в най-лошия случай симулацията обхожда всеки кораб, то тя трябва да е със сложност O(N * M). И тъй като я пускаме N * M пъти, то цялата сложност на това решение би била $O(N^2 * M^2)$. За простота, ако считем, че N ~= M, сложността е $O(N^4)$. Това е и сложността, която се искаше в задачата.

Забелязахте ли грешката в горния параграф? Не е много очевидно (а и на всичкото отгоре не се показва на рандом тестове), че симулацията всъщност не е O(N * M), ами O(N * M * max(N, M)). Защо? Нали казахме, че в най-лошия случай обхождаме всеки кораб точно по веднъж? Да, само че нищо не сме казали за всяка празна клетка! Представете си например следната дъска с един ред и 50 колони:

>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<

Колко би продължила най-дългата игра тук? Оказва се, че цели 1275 хода (ако почнем в някоя от двете "средни" клетки). Което е доста повече от 50, и, както можете да видите, всъщност е около N^2 / 2.

Съответно, могат да се създадат тестове, в които игрите да продължават около N^2 * M или N * M^2 хода. Нещо повече, с повече креативност могат да се създадат тестове, в които игрите продължават по (що-годе) толкова *независимо от коя клетка тръгнем*! Което от своя страна означава, че цялостното решение става $O(N^2 * M^2 * max(N, M))$, или, ако считем, че N ~= M, $O(N^5)$.

Добре, тогава как да стигнем до $O(N^4)$, която казахме, че искаме? Ами, ще приложим нещо, което много рядко *реално* се ползва в състезателни задачи – именно свързани списъци! За всяка клетка от дъската ще си пазим нейните четири съседа – нагоре, надясно, надолу и наляво. Когато ударим кораб в дадена клетка, ние ще знаем (и ще можем да достъпим константно) всеки от четирите му съседни кораба, игнорирайки празните пространства. Така ние ще можем да ъпдейтнем техните съседи по такъв начин, че да отчетат премахването на кораба от текущата клетка.

Нещо повече, по този начин, знаейки, че се намираме в даден кораб (неговата начална клетка), ние можем за O(1) да стигнем до следващия кораб (или да видим, че излизаме извън дъската) *независимо* каква е дъската! Така вече симулацията ще стане O(N * M), и решението ни наистина ще бъде $O(N^2 * M^2)$.

Тъй като не исках да бъда прекалено груб (тайм лимитите са настройвани по решението ми, което не само ползва оптимизацията със свързаните списъци, ами е и относително ефективно), съм дал само 4 теста (даващи 20 точки), които да са от "гадния" тип. Така предполагам повечето състезатели ще хванат около 80 точки с най-простото решение, което пробва всяка начална клетка и симулира играта (което, само по себе си, е адекватна задача за тази група). Останалите 20 точки са за хардкор състезателите :)

Автор: Александър Георгиев