АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
FUEL

(решение)

Задачата е третата от поредицата „Fuel“, но всяка от тях има различно решение (а и другите са давани на студентски състезания). Един hint към младите състезатели, който мога да дам (и ще им е много полезен в бъдеще) е когато в някоя задача се пита „по колко начина“, то тя с над 80% вероятност се решава с динамично оптимиране. Тази не е изключение от това правило.

Най-простото решение би било да пуснем пълно изчерпване, което да намира отговора на задачата. То, за съжаление, е линейно спрямо отговора, а както е казано в самото условие, той може да бъде много голям. Всъщност както отговорът, така и времето за изпълнение на това решение са експоненциални, а тъй като N е голямо, това не е добре за нас. Това решение, макар и просто, би хванало сравнително малко точки.

Ако все пак състезателите са се сетили за динамично оптимиране, то първата им идея най-вероятно би била да пазят в стейта на динамичното до къде е стигнала Ели (тъй като независимо как тя е стигнала до там, броят начини да стигне от там до края е един и същ). За всеки стейт те трябва да определят следващата бензиностанция, на която тя ще спре, което е цикъл с дължина до N. Тъй като броят състояния е O(N), и вътрешно имаме цикъл със сложност O(N), то общата сложност на това решение би била O(N*N). Това решение е значително по-добро от предходното, и съответно тестовете са така предвидени, че то да хване около половината точки.

Решението за 100 точки де факто е същото като предходното, като правим най-простия вариант на „оптимизация на вътрешния цикъл“, която... еми премахва вътрешния цикъл. При итеративната реализация на динамичното, решението би било да тръгнем от последната клетка (където е университета) и да кажем, че от там до края можем да стигнем по един единствен начин (това си е така). После, от предпоследната клетка сумираме всички следващи клетки (които са вече изчислени, в случая само последната), стига те да са на разстояние по-малко или равно на максималното, което Ели може да измине. Така намираме резултата за клетка N-1 (ако в N включим началната и крайната позиция). За позиция N-2 отново сумираме няколко от следващите клетки (в случая най-много N-1 и N). Обобщавайки, за позиция P, ние сумираме всички клетки на позиции P+1, P+2, …, P+k, където позицията на клетка P+k все още е по-малка или равна на L, а тази с P+k+1 е по-голяма (или невалидна, ако P+k == N). Какво можем да видим за последната клетка от тази сума? Тя винаги или остава същата като предходната, или намалява. Тоест функцията на последната клетка от сумата е монотонна. Това не би трябвало да ви говори много, затова ето пример. Ако за клетка P последната клетка от сумата е била P + k, то за клетка P-1 последната клетка НЕ би могла да бъде P + k + някакво ненулево число (тоест по-голяма от P + k). Следователно, вместо да въртим цикъл отново и отново, в общия случай минавайки през едни и същи клетки по много пъти, ние просто можем да пазим една обща сума, като на всяка стъпка я увеличаваме с текущата клетка, и евентуално я намаляме преди разглеждането на следващата стойност от динамичното, ако се

появят нови недостижими клетки (както казахме, ако те са недостижими за текущата клетка, то те биха били недостижими и за всички следващи клетки, което следва от монотонността на функцията).
На пръв поглед вътрешен цикъл все още има (който намира колко клетки стават недостижими при започване разглеждането на всяка от стойностите на динамичното). Той, обаче, ще премине през всяка клетка по най-много веднъж. Тоест сложността на цикъла отново е O(N), но този път тя е ИЗВЪН сложността на самото динамично. Така решението става O(N + N), тоест линейно, вместо досегашното O(N * N), което беше квадратно.

Автор: Александър Георгиев