

# АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПИРАТИ

## Решение 1:

Очевидно задачата може да се сведе до такава за разширяване на графа. Един връх от графа ще ни бъде кодиран със следните параметри  $x, y, t$ , където  $x$  и  $y$  са координатите на клетка от картата, а  $t$  е времето в което сме стигнали в тази клетка. След което просто трябва да се имплементира алгоритъм на Дейкстра в така зададения граф и да се намери минималния път като за цена на теглата се използва горивото за преминаване между 2 клетки. След като сме направили горният анализ остава малко по-трудната задача, а именно: "Как за дадена клетка  $x, y$  и време  $t$  да се намери регионът, който ни трябва?". Решенията на това са няколко, за тестовите с помалки ограничения може да се използва някакво преизчисление по обновлението на данните, но за наистина тежките тестове трябва да се използва 2D интервално дърво. В клетките на това дърво ще се пази сортиран вектор по време от регионите, които са били добавени за тази клетка  $x, y$ . Сложността на заявката по време за намиране на искания регион е  $\log(N) * \log(M) * \log(r)$ , където  $r$  е на броят на добавените региони в клетката. От там цялостната сложност за задачата е  $N * M * D * \log(N) * \log(M) * \log(r)$ . Паметта, от която имаме нужда за дървото е  $4 * N * M + 2 * \log(N) * \log(M) * \log(ar)$ , Където  $ar$  е броят на по-късно добавените региони в задачата.

## Решение 2:

С малко повече вглеждане можем да забележим следната релация за решение на задачата:

$$f(x, y, time) = \min( f(x - 1, y, time - 1) + fuel1, \\ f(x + 1, y, time - 1) + fuel2, \\ f(x, y - 1, time - 1) + fuel3, \\ f(x, y + 1, time - 1) + fuel4 )$$

Тази релация ни дава възможност да напишем решение използвайки тримерно динамично. Както се вижда ни трябва само последната таблица от динамичното. Освен това понеже обновлението ни са сортирани във възходящ ред можем да правим итеративно обхождане на таблицата за всяко  $t$  и да проверяваме дали имаме обновление, ако имаме изчитаме го и обновяваме таблицата `active_region`, където пазим само активните региони. Сложността на решението става  $M * N * D$  и изискваната памет е  $4 * M * N$ . Към горната идея са добавени още 2 оптимизации: С първата се лишаваме от нуждата на `memset` и използваме 2 `pointer`-а, които да разменяме всеки път когато изчисляваме следващата стъпка от таблицата. С използването на битова карта с големина 128 на 128 можем да се лишим от `memset` операцията на стъпката която ще изчисляваме, по този начин можем директно в цикъла на динамичното да решим дали трябва да сложим максимална стойност на клетката.

Авторовото решение използва решение 2.

*Автори: Господин Бодуров,  
Иван Георгиев,  
Николай Хубанов*