АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА **PARK**

Прави впечатление, че в задачата възможните стойности за K са от 1 до 109. Това означава, че трябва да намерим начин, по който да „изобразим“ всяко число в този интервал.

Една от първите идеи, която може да ни дойде на ум, е да представим числото като сума от биномни коефициенти, тъй като и самата задача за намирането на броя на най-кратките пътища между 2 срещуположни ъгъла в правоъгълна таблица е тясно свързана с тях[[1]](#one). Този подход, макар и да дава верни резултати, е доста бавен в някои случаи, защото нямаме точно дефиниран алгоритъм, по който можем да изградим всяко число.

Един от основните начини за представяне на числа е двоичната бройна система. За разлика от идеята с биномните коефициенти, тук вече имаме алгоритъм, по който да представим всяко едно число – като използваме степените на двойката, участващи в двоичния му запис. Обаче, нямаме ясно дефиниран начин, по който да изобразим тези степени в таблицата (паркът).

|  |  |
| --- | --- |
| Нека разгледаме следната таблица (*табл. 1*).  В нея има точно 2 пътя от горния ляв край до долния десен. | *Табл. 1* |
| Сега нека „слепим“ две такива таблици (маркирани в червено и синьо) по следния начин (*табл. 2*).  Можем да намерим броя на пътищата като за всяка клетка добавим броя на пътищата до съседната ѝ вляво и до съседната ѝ отгоре. До началната клетка има само 1 път. За последната този брой е 4 (*табл. 3*). | *Табл. 2*  *Табл. 3* |
| Продължавайки с тази конструкция, получаваме (*табл. 4*).  Броят на пътищата е 8 (*табл. 5*). | *Табл. 5*  *Табл. 4* |
| Ако продължим със същата идея ще видим, че добавяйки нова таблица ще удвоим пътищата до най-долната дясна клетка (която ще е добавена на този ход).  Това може лесно да се докаже (*табл. 6*), където в червено е оказана най-долната таблица от предишната конструкция, докато в синьо – таблицата която сме добавили. | *Табл. 6* |

Всъщност, горните наблюдения се дължат на следния по-общ факт:

|  |  |
| --- | --- |
| I. Ако имаме две таблици, като в първата броят на пътищата е ***А***, а във втората – ***Б***, и ги слепим така, че долния десен край на първата да съвпада с долния десен на втората, то общия брой пътища в новополучената таблица ще е ***А\*Б*** (*табл. 7*). | *Табл. 7* |

Друг по-общ факт е:

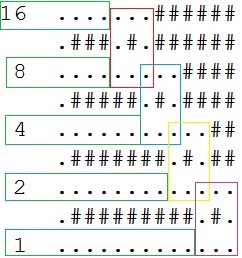
|  |  |
| --- | --- |
| II. Ако имаме две таблици, като в първата броят на пътищата е ***А***, а във втората – ***Б***, и ги свържем чрез пътища така, че едната да е строго по-нагоре и по-надясно от другата, то общия брой пътища до най-долната дясна клетка в новополучената таблица ще е ***А+Б*** (*табл. 8*). В зелено са означени пътища, по които може да се преминава по единствен начин. | *Табл. 8* |

Така вече се научихме как да можем да представим таблица, в която броят на пътищата е някаква степен на двойката. Обаче за да представим едно число по този начин ще са ни нужни няколко такива конструкции за представяне на всяка степен на двойката. Поради ограниченията в задачата (максимално допустимата таблица, която може да конструираме е с размери 200 на 200) е възможно тези наблюдения да са ни достатъчни за решението ѝ – генерираме такива конструкции за всички степени на двойката, които се срещат в двоичния запис на числото и впоследствие ги свързваме по подходящ начин, използвайки факт **II**.

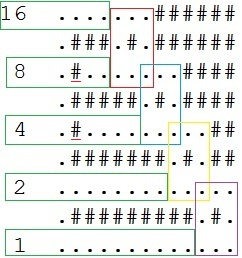
Но нека се пробваме да използваме една и съща конструкция за това. Нека тази конструкция е изградена от *N* на брой таблици, всяка от които като *табл. 1*. Броят на пътищата, започвайки от най-горния ляв ъгъл на първата такава таблица ще е 2**N**. Ако започнем от най-горната лява клетка на втората таблица, то броят на пътищата ще е 2**N-1**. И така, ако започнем от *i-тата* таблица от конструкцията, броят на пътищата до най-долната дясна клетка на последната от таблиците ще е 2**N –i +1**.

|  |  |
| --- | --- |
| Проблемът е, че ако просто построим пътища до тези клетки, то те биха ни развалили конструкцията. Примерно, ако искаме да има 10 пътища, направени с помощта на конструкцията, то трябва клетките, маркирани в зелено да ги направим допустими за преминаване. Тогава обаче няма да получим желаните 10 пътища, а по-голяма бройка, тъй като идеята на конструкцията се нарушава. | *Табл. 9* |
| Решението е лесно – вместо да използваме таблицата показана чрез *табл. 1*, ще използваме следната (*табл. 10*), в която броят на пътищата е отново 2, но вече може да добавяме пътища без да разваляме конструкцията. | *Табл. 10* |
| Сега нека отново да се пробваме да построим 10 пътища, но вече използвайки *табл. 10* за новата конструкция (*табл. 11*). Таблиците от *табл. 11*, означени с даден цвят съвпадат със съответната таблица от *табл. 9*, маркирана със същия цвят.  Ако заменим клетките, маркирани в зелено с допустими за преминаване, то от долната група ще имаме 2 различни пътя, докато от горната – 8 различни пътища, или общо 10, колкото ни и трябваха. | *Табл. 11* |

Сега остава само да построим цялостното решение (като пример за цялата конструкция ще дадем таблица, даваща решение за всяко *K* от 1 до 31):



Вляво е означен броят на различните пътища до най-долната дясна клетка, ако използваме пътя, маркиран в зелено. Така както е дадена, таблицата решава случая, в който K=31. За да представим някое произволно число до 31, трябва да забраним пътищата някои от пътищата, маркирани в зелено (и по-конкретно тези, чийто брой пътища не участва в двоичното представяне на числото). Така, ако искаме да имаме 19 различни пътища от най-горната лява до най-долната дясна клетка, то трябва да разгледаме двоичния му код. Той е 10011 и следователно числото може да се представи като 16 + 2 + 1. Остава и да забраним всички пътища в зелено, които не отговарят на някое от тези числа:



Конструкцията, която решава цялата задача (за K от 1 до 109) се побира в таблица с размери 59 на 61. В реализацията трябва веднъж да построим описаната конструкция и след това да разгледаме двоичния запис на числото (сложност log n) и да маркираме някои от клетките. Това е и цялостната сложност на решението.

*Кратък коментар:*

Въпреки, че в голяма част от състезанието имаше проблеми с тестването на задачата, множество състезатели изпратиха решения. От тях обаче само 3 са верни – на Румен Христов, Христо Венев и Веселин Георгиев (чийто бърз събмит на тази задача му подсигури победата, дори и Владислав Харалампиев да беше решил и 5-тата задача).

*автор: Ясен Трифонов*

**Препратки:**

[1] <http://joaoff.com/2008/01/20/a-square-grid-path-problem/>