

## Обяснение на задача – „Кой е бащата“

В задачата се търси дали даден низ (дете) е комбинация от подредици на два други низа (родители). Задачата се решава като съобразяваме че всеки знак от детето трябва да идва от един от двата родителя, и че знаците от всеки родител спазват оригиналната си подредба. За да открием знаците на детето в родителите можем да поставим по един указател в началото всеки низ и да сканираме отляво надясно. По този начин стигаме до рекурсивно решение – ако указателят на детето е `indexTarget`, а на родителите `indexA` и `indexB`, то изглежда така:

- Ако `indexTarget` успешно стигне до края на детето, значи сме намерили всички знаци от родителите, и отговорът е „Да“, т.е. детето наистина е на тези родители (базов случай).
- Иначе, ако `indexA` и `indexB` стигнат до края на родителите, значи сме изчерпали възможностите и отговорът е „Не“, поне за тази комбинация от движения която сме извършили с указателите (базов случай).
- Иначе, ако знакът до който сме стигнали в детето съвпада със знака в един от родителите, можем да преместим указателя в детето и съответният родител (рекурсивно извикване). Така решаваме че даденият знак в детето идва от този знак в родителя.
- Независимо от това дали знакът съвпада, можем да преместим единият родителски указател без да пипаме този в детето (рекурсивно извикване). Така решаваме че даденият знак в родителя не участва в детето.

Общо имаме 4 рекурсивни извиквания и експоненциален растеж на дървото на рекурсия. Това може да се избегне като приложим динамично оптимизиране:

- Проблемът е рекурсивен, защото с всяко местене на указател получаваме по-малка инстанция на същия проблем.
- Проблемът има оптимална субструктура защото крайният резултат е просто логическо „или“ на всички базови случаи на рекурсията.
- Проблемът преизчислява едни и същи стойности тъй като имаме три аргумента за състояние (позициите на трите указателя), т.е.  $A*B*C$  възможни извиквания, където  $A$ ,  $B$  и  $C$  да дължините на трите низа, докато броят извиквания е експоненциален.

За да стигнем до алгоритъма с динамично оптимизиране, обръщаме посоката на рекурсия като движим указателите отляво наляво (от  $A, B, C$  до  $0, 0, 0$ ). Таблицата ни е триизмерна заради трите аргумента, с булеви стойности, и описва дали в дадена комбинация от позиции на указателите, отговорът е положителен. Базовите случаи са в клетките с последната позиция детския указател, където сме изчерпали детето и знаем че отговорът е „Да“ – за да стигнем до решението обхождаме таблицата от  $[A, B, C]$  до  $[0, 0, 0]$  като попълваме клетките и търсим дали има пътека от клетки с „Да“ която да стига до началото на таблицата (за справка вж. авторското решение). Попълването на клетките е аналогично на рекурсивните извиквания, като проверяваме дали знакът в детето при тези позиции на указателите съвпада с някой от родителите.

Забележка: В авторското решение, рекурсивният алгоритъм има 7 рекурсивни извиквания, като някои не са необходими защото се покриват от останалите случаи. Кои са те?