АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПЪЗЕЛ

Първата задача може да се реши лесно с подхода на динамичното програмиране. Може просто да се направи линейно обхождане по номера изписан върху парчетата. Когато сме стигнали до номер *i*, пазим по колко начина може да наредим пъзел от първите i парчета от пъзела (с номера от 1 до *i*), завършвайки на всеки от 3-те възможни краища. След като вече сме сметнали за първите *i* парчета, лесно можем да определим колко са начините за *i*+1 парчета. Това ще стане като към всеки от 3-те краища на наредбите на първите *i* парчета, долепим парче с номер *i*+1 и така получаваме броя начини да наредим първите *i*+1 парчета завършващи на всеки от 3-те краища. По този начин, накрая получаваме отговора като стигнем до парчетата с номер *n* и в частност ни интересуват възможните наредби на първите *n* парчета завършващи на край от тип 1.

Втората задача е по-интересна. Често, когато към задача, която се решава с динамично, добавим заявки за промяна на някакви елементи в нея, задачата може да се реши използвайки някакво дърво в което да пазим същите неща които пазим в динамичното. Тази задача е точно такава.

Идеята е да поддържаме индексно дърво над номерата на парчетата от пъзела, във върховете на което да пазим следната информация: По-колко начина можем да наредим парчетата с номера от *i* до *j* завършвайки отляво на край от тип l и отдясно на край от тип *r*. Където [*i, j*] е интервала от индекси за които отговаря върха в дървото, а броят начини пазим за всяка двойка l и *r* (общо 9 двойки).

Така, алгоритъмът се свежда до това за всяка заявка да променяме някое листо (а именно листото отговарящо на номера на парчета което добавяме/махаме). После минаваме по пътя от листото до корена и преизчисляваме за всеки връх по пътя броя начини (използвайки данните във левия и десния преки наследници). Това довежда до логаритмичен брой операции за всяка заявка. Накрая отговорът се крие в корена на дървото (в частност броя начини за завършване отляво и отдясно на край от тип 1). Удобно е първите M парчета да ги третираме просто като заявки за добавяне. Така общата сложност на алгоритъма е $O((M+Q)\log N)$.

Автори на анализа: Иво Дилов, Даниел Атанасов

Автор на решенията: Иво Дилов