

Решение на задача Тото:

Приготвил: Веселин Георгиев

Анализ

Задачата допуска няколко решения - всички са базирани на някакво *търсене*. Доброто решение, което е предвидено да хване 100 точки се базира на двоично търсене. Линеинo търсене със сигурност не хваща всичките точки - колко точно ще получи зависи от конкретната реализация.

Разгледани са следните алгоритми:

Алгоритъм 1 ("тъпо" линеинo търсене):

Пробват се комбинациите [1, 2, 3, 4, 5, 6], [2, 3, 4, 5, 6, 7] и така нататък докато `post_comb` не върне 0 (т.е. не получим група от 6 числа, от които никое не е от комбинацията на журито).

Нека комбинацията $[x, x+1, x+2, x+3, x+4, x+5]$ дава 0.

Тогава, за всяко i , такова, че $i < x$ и $i > x+5$ се пуска комбинацията $[i, x, x+1, x+2, x+3, x+4, x+5]$; ако `post_comb` върне 1, значи числото i участва в решението.

Алгоритъм1 получава около **25** точки.

Алгоритъм 2 ("по-умно" линеинo търсене):

Той се различава от Алгоритъм 1 по началното търсене на "празна" шесторка: пробват се [1, 2, 3, 4, 5, 6], [7, 8, 9, 10, 11, 12] и така нататък - съгласно принципа на Дирихле, празна шесторка ще бъде намерена, и то най-много за 7 викания на `post_comb`. Оттук нататък, алгоритъма съвпада с Алгоритъм1.

Алгоритъма се оценява на около **35** точки.

Други реализации на линеинo търсене: В зависимост от конкретната реализация и включването на някои хитрости, е възможно да се постигнат 60-75 точки.

Алгоритъм 3 (авторовото решение - двоично търсене):

Подобно на Алгоритъм 2, пробват се 8-те "слота" [1, 2, 3, 4, 5, 6], [7, 8, 9, 10, 11, 12] ... [43, 44, 45, 46, 47, 48]. Поне два от тези слотове ще са празни. Освен това, ако сумата на намерените числа в слотовете е 5, значи числото 49 участва в решението. В този момент фиксираме един от празните слотове, който ще използваме да "запълваме" запитванията оттук нататък.

За всеки слот, в който има точно едно число правим двоично търсене. Това става така. Нека например знаем, че измежду числата [1, 2, 3, 4, 5, 6] има едно число от комбинацията на журито. Нека имаме и че числата 7, 8, 9, 10, 11, 12 са "чисти".

Първият ни въпрос е "1, 2, 3, 7, 8, 9", т.е. питаме за три от числата от въпросния слот, като допълваме въпроса с три числа от някой свободен слот. Ако `post_comb` върне 1, значи ще продължим търсенето в първата половина на слота [1, 2, 3, 4, 5, 6], инак – във втората половина. Нека се е оказало, че `post_comb` е върнало 0. Пускаме въпрос "4, 7, 8, 9, 10, 11". При 1 – числото 4 участва в решението и прекратяваме търсенето. При нула питаме за "5, 7, 8, 9, 10, 11" и при 1 заключаваме, че 5 участва в решението, инак 6 участва в решението. Така можем да се убедим, че да намерим число от даден слот става със задаване на най-много 3 въпроса. Това ни дава абсолютен максимум на питанията $6 \cdot 3 + 8$ (осмицата е от първоначалните питания за свободен слот), което е **26 въпроса**. За съжаление ни трябва и 27-ми въпрос, за да дадем отговора.

Какво правим, когато имаме две числа в един и същи слот? Тъй като с двоично търсене теоретически ни трябват $2 \cdot 3 = 6$ питания, а с директно (линейно) търсене – пак 6 (а може и 5, ако се възползваме от информацията до момента – т.е. ако сме питали за първите 4 числа и никое не е решение, то последните 2 в слота очевидно участват). Разбира се, двоично търсене (с леки модификации) също става (и на практика се държи малко по-добре).

В задачата са разрешени 28 питания – в случай, че някой не се сети за "хитростта" с 49-тото число.

"Най-добро решение":

Възможно е (но не и в зададеното време) задачата да се реши със значително по-малко опити (около 18 в най-лошия случай!).