Şir

Soluţie (Mircea Dima)

Vom calcula un vector ***next[i]*** care ne va spune poziţia următorului element care are valoarea ***S - a[i]***. În caz că nu există o astfel de poziţie, setăm ***next[i] = -1***. Acesta se calculeaza în ***O(n)*** folosind un *hash*.
Acum, ne rămâne să verificăm pentru fiecare interval ***[i, next[i]]*** dacă respectă proprietăţile. Acest lucru se poate face liniar obţinându-se ***$O(n^2)$*** **(40 pcte)** sau putem considera puncte în plan de coordonate ***(i, next[i])***. Observăm că o subsecvenţă ***[i, j]*** respectă cerinţele dacă dreptunghiul ***[i,i, j,j]*** conţine exact ***(j-i+1)/2*** puncte.
Se poate implementa un arbore de intervale obţinându-se complexitate
$O(n \log^2 n)$ sau ***O(n log n)*** folosind rafinamentul Willard-Lueker. **(100 pcte)**

Soluţie 2

Pornim de la observaţia că dacă o valoare ***x*** aparţine şirului, valoarea ***S – x*** ori nu există, ori este unică. Vom calcula un vector ***other[i]*** care reţine poziţia elementului ***S – a[i]***, dacă există. În continuare, pentru fiecare poziţie din şirul numerelor vom găsi un interval ***[min(i, other[i]), max(i, other[i])]***. Intervalul respectă condiţia întâi, iar pentru a respecta şi a doua cerinţă va trebui ca cea mai mare şi cea mai mică valoare a vectorului ***other[]*** din acest interval să nu depăşească marginile intervalului. Determinarea valorii maxime şi minime pe un interval se poate determina în complexitate ***O(log N)*** cu arbori de intervale. Complexitatea finală: ***O(N log N)***.