Zadanie: ZEP
Zerowe przedziały

Warsztaty ILO 2017-2018, grupa olimpijska. Dostępna pamięć: 128 MB.

Niech $S = a_1 + a_2 + \cdots + a_n$.

Rozwiązanie wzorcowe $O(nS)$

Zadanie rozwiążemy metodą programowania dynamicznego. Będziemy obliczać tablicę $DP[n][S]$, która dla każdego prefiksu ciągu $i \leq n$ i każdej sumy $j \leq S$ pamięta, ile jest poprawnych konfiguracji, takich że przedziały kończą się na pozycji i a elementy sumują się do j.

Obliczamy to dla danego prefiksu w następujący sposób. Mamy dwa przypadki:

1. Ostatni element (i-ty) doklejamy do jakiegoś przedziału kończącego się na pozycji $i - 1$. Wówczas $DP[i][j] = DP[i-1][j-a_i] + DP[i-1][j+a_i]$, ponieważ musimy rozważyć dwie konfiguracje – dany element ma znak dodatni lub ujemny.
2. Tworzymy nowy przedział, w którym jest tylko element i-ty. Wówczas zwiększamy $DP[i][a_i]$ oraz $DP[i][-a_i]$ o 1.

Rozwiązaniem zadania jest suma po wszystkich i z $DP[i][0]$, ponieważ pozwalamy na kończenie się przedziału w dowolnym miejscu i chcemy mieć sumę równą 0.

Złożoność wynosi nS, ponieważ mamy nS stanów, a dla każdego stanu stałą liczbę przejść do innych stanów.

Należy tu uważać a ujemne liczby, ponieważ indeksy tablicy mogą również być ujemne. Wystarczy w tym celu 'przesunąć' wszystkie indeksy o pewną stałą tak, aby najmniejsza możliwa suma po przesunięciu była dodatnia.

Rozwiązanie alternatywne $O(nS\ log\ n)$

To rozwiązanie korzysta z techniki *dziel i zwyciężaj*. Dzielimy nasz ciąg na pół. Rekurencyjnie obliczamy poprawne konfiguracje w lewej połowie ciągu oraz w prawej połowie ciągu. Następnie musimy jeszcze obliczyć konfiguracje takie, że przedział zaczyna się w lewej połowie, a kończy w prawej. W tym celu obliczymy dla każdego sufiksu lewej połowy i dla każdego prefiksu prawej połowy obliczymy wszystkie możliwe konfiguracje sumujące się do każdej możliwej wartości j.

Tutaj również użyjemy programowania dynamicznego. $DPsuf[i][j]$ pamięta ile jest konfiguracji, które są sufiksem o długości i lewej połowy i wartości po określeniu znaku sumują się do j. Łatwo zauważyć, że $DPsuf[i][j] = DPsuf[i-1][j - suf_i] + DPsuf[i-1][j + suf_i]$, gdzie suf_i jest i-tym elementem sufiksu. Analogicznie obliczamy $DPpref[i][j]$ dla prefiksów prawej połowy.

Następnie chcemy policzyć konfiguracje zaczynające się w lewej połowie, kończące się w prawej połowie, które sumują się do 0. Zatem dla dowolnej wartości x zsumujmy konfiguracje sumujące się do x w lewej połowie oraz konfiguracje sumujące się do $-x$ w prawej połowie. Następnie wziąć dowolną taką konfigurację z lewej połowy i połączyć ją z dowolną konfigurację z prawej połowy i suma wyniesie 0. Zatem przemnażamy obliczone wyniki i sumujemy wyniki po wszystkich x.

Tutaj również należy uważać na ujemne indeksy.