

Opis działania

Trzymamy osobno przedziały gdzie są 0 i 1 na drzewach AVL posortowanych po pierwszym elemencie przedziału.

Ini:

Dzielimy ciąg na spójne przedziały. Następnie z tych przedziałów tworzymy drzewo.

Zam(i):

1. Szukamy przedziału na drzewach AVL, w którym jest ten element. Robimy to szukając największego pierwszego elementu mniejszego bądź równego i i patrząc czy w tym przedziale jest i .
2. Usuwamy ten przedział z odpowiedniego drzewa. Niech ten przedział to $[a, b]$. Wsadzamy wtedy o ile nie są puste i poprawne przedziały $[a, i - 1]$ i $[i + 1, b]$ z powrotem do tego drzewa, z którego wyjeliśmy i .
3. Szukamy przedziałów $[c, i - 1]$ i $[i + 1, b]$ na drzewie AVL, w którym nie było elementu i . Przedziału $[c, i - 1]$ szukamy szukając największego pierwszego elementu mniejszego bądź równego i i patrząc czy ten przedział kończy się na $i - 1$. Szukanie przedziału $[i + 1, b]$ jest trywialne jako że pierwszy element jest kluczem w drzewie. Usuwamy te przedziały o ile je znaleźliśmy. Złączamy znalezione przedziały ze sobą i z i i wkładamy do drzewa, w którym nie było elementu i .

Blok(i):

Szukamy w analogiczny sposób do $Zam(i)$ przedziału, na którym znajdują się i i zwracamy jego długość.

Złożność:

Ini - $O(n)$

Złożoność jest $O(n)$ a nie $O(n \log(n))$ ponieważ początkowa lista przedziałów jest już posortowana, więc możemy w prosty sposób zbudować drzewo od razu podłączając kolejny element w odpowiednie miejsce nie szukając go wcześniej.

Zam - $O(\log(n))$

Wyszukania i włożenia na drzewie AVL.

Najbliższa - $O(\log(n))$

Wyszukanie na drzewie AVL.