Examen Structuri de Date

Subiectul 2

1) Se dă o matrice n*n și m operații de 2 tipuri

- Adaugă x elementului de pe poziția i j
- Care este a k-a cea mai mare linie din punct de vedere al sumei.

1.5p

Rezolvare:

Arbori binari de căutare

Calculăm suma pentru fiecare linie a matricei. Inserăm într-un arbore binar de căutare echilibrat suma corespunzătoare fiecărei linii, și balansăm arborele corespunzător. Pentru operația de tipul 1 vom adăuga la suma care corespunde indicelui i valoarea x (chiar dacă avem o matrice, valoarea adăugată la un singur element va modifica suma întregii linii). Această operație se efectuează în O(logn). Pentru operația de tipul 2 vom recalcula maximul pe care îl vom memora într-o variabilă auxiliară. Parcurgem arborele în inordine (SRD)și actualizăm maximul. Această operație se efectuează în O(logn), deoarece căutăm al k-lea maxim într-un arbore binar de căutare echilibrat.

https://www.geeksforgeeks.org/find-k-th-smallest-element-in-bst-order-statistics-in-bst/

2) Se dau n valori. Calculați un vector cu alte n valori care sunt obținute din scăderea fiecarui element a primului element din dreapta mai mic ca el, dacă există.

Exemplu [9,4,3,6,1] -> [5,1,2,5,1]

5 = 9 – 4

1 = 4 - 3

3 = 2 - 1

5 = 6 - 1

1p

Rezolvare:

Stivă

Introducem primul element într-o stivă. Iterăm prin valorile rămase aplicând următorul algoritm: dacă stiva nu este vidă, atunci comparăm valoarea curentă cu vârful stivei. Dacă aceasta este mai mică, atunci facem diferența dintre cele două și o introducem în vector. Eliminăm elemente din stivă și repetăm procedeul până când valoarea curentă este mai mare decât vârful stivei sau stiva este vidă. Atunci introducem valoarea curentă în stivă. Repetăm procedeul până când lista de elemente citită este nulă. Complexitatea este de O(n), timp liniar.

3) Se dau 2 vectori de n elemente. Amandoi sortați. Găsiți al k-lea cel mai mic element din reuniunea lor. Explicați algoritmul + complexitatea.

1p

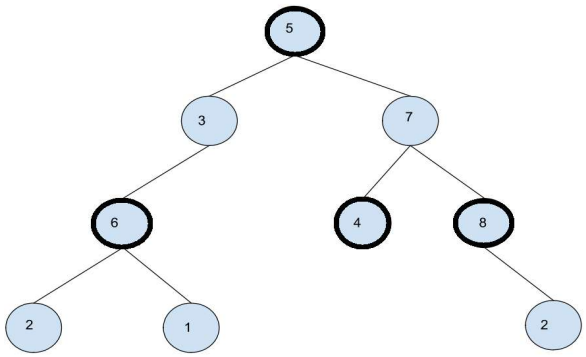
Rezolvare:

Căutare Binară

Împărțim cei doi vectori în subvectori de dimensiune k / 2. Pentru fiecare element de pe ultima poziție verificăm în ce relație se află cu cel din celălalt vector. Adică, dacă v1[i] > v2[j], atunci înseamnă că am luat prea multe valori din primul vector și trebuie să restrângem intervalul. Ne oprim dacă elementul maxim pe care îl luam din primul vector este egal cu elementul maxim pe care îl luăm din cel de al doilea vector sau dacă cel mai mare element pe care îl luăm din primul vector este mai mic decât cel mai mic element pe care nu îl luăm din cel de-al doilea vector sau dacă cel mai mare element pe care îl luăm din cel de-al doilea vector este mai mic decât cel mai mic element pe care nu îl luăm din primul vector. Complexitatea este de O(logk)

https://www.geeksforgeeks.org/k-th-element-two-sorted-arrays/

4) Se dă un arbore binar, găsiti suma maxima a unor elemente care nu se învecinează.



În acest exemplu suma maximă e 23.

Explicați cum functioneaza algoritmul pentru a gasi suma maxima si care este complexitatea sa.

1p

Rezolvare:

Programare Dinamică

Stocăm suma maximă incluzând valoarea nodului current sau excluzând-o, dacă este adiacentă. Se efectuează apel recursiv pentru nepoții nodurilor care sunt incluse sau simplu apel pentru vecini, în cazul în care nodul este exclus. Complexitate O(n).

https://www.geeksforgeeks.org/maximum-sum-nodes-binary-tree-no-two-adjacent/