

Seksjon 1

1 OPPGAVE

1. (25%)

Skriv (i Java plus standard library, eller i pseudokode) en metode

`int threeSumNeg(int[] a)`

som returnerer antallet tripler i, j, k med $0 \leq i < j < k < a.length$ slik at $a[i] + a[j] + a[k] < 0$. Du kan anta at alle elementer i tabellen a er forskjellige. For å oppnå full skåre bør din metode kjøre i $O(N^2 \log N)$ tid, der $N = a.length$.

Fill in your answer here

2 OPPGAVE

2. (25%)

La `int[] a` være en tabell av heltall mellom 0 og 99. En slik tabell kan bli sortert effektivt gjennom å bruke en hjelpetabell `int[100] count`. Skriv en metode (i Java eller pseudokode, uten å bruke en metode som `sort()` fra et bibliotek)

`void sort(int[] a)`

som sorterer a . For å oppnå full skåre bør din metode kjøre i $O(N)$ tid, der $N = a.length$.

Fill in your answer here

3 OPPGAVE

3. (25%)

Forklar hva en "hash table" er. Beskriv først virkemåten ved å forklare hva de metodene `get()`, `put()` og `delete()` i API-en bør gjøre. Forklar så hva en "hash function" er og hvorfor "collision resolution"

er viktig. Beskriv til slutt en implementering ved å gi passende datastrukturer samt en beskrivelse av en implementering av de tre ovennevnte metodene.

Fill in your answer here

4 OPPGAVE

4. (25%)

Gitt en rettet graf G med noder $0, \dots, V-1$, representert vha nabolister av ut-piler. Mer presist, for enhver node v , er $adj[v]$ en kjedede liste som inneholder alle noder w slik at G har en pil fra v til w . Skriv (i Java eller i pseudokode) en metode

boolean reachable(int s, int t)

som returnerer en boolsk verdi *true* hvis og bare hvis det er en sti fra s til t i G .

(Hint: ingen informasjon om selve stien (hvis den finnes) trengs å bli returnert.)

Analysér kjøretiden i verste fall av algoritmen din i termer av antall noder og piler i G .

Fill in your answer here