

582359 Algoritmit ongelmanratkaisussa (kevät 2013)

Viikon 5 tehtävät (15.2.)

1. Toteuta yksiulotteinen summataulukko ja \sqrt{n} -rakenne. Tee testi, jossa taulukossa on 1000000 lukua ja siitä lasketaan 50000 summaa satunnaisilta väleiltä. Paljonko menee aikaa, kun summat lasketaan suoraan taulukosta for-silmukalla, summataulukosta ja \sqrt{n} -rakenteesta?
2. Toteuta kaksiulotteinen summataulukko ja testaa sen toimivuus.
Hahmottele, miten kolmiulotteisen summataulukon avulla voisi laskea kolmiulotteisen taulukon minkä tahansa kolmiulotteisen alitaulukon lukujen summan ajassa $O(1)$.
3. Lähtökohtana on seuraava taulukko:

9	1	4	9	8	5	5	6
---	---	---	---	---	---	---	---

- (a) Piirrä segmenttipuu, jonka avulla voi laskea taulukon välien summia.
 - (b) Näytä, miten segmenttipuusta saadaan laskettua summa luvusta 4 lukuun 8 ja luvusta 1 lukuun 6.
 - (c) Näytä segmenttipuun muutokset, kun luvun 4 tilalle muutetaan 5 ja luvun 6 tilalle muutetaan 3.
 - (d) Toista kohdat (a)–(c) niin, että summien sijasta kyselyt antavat pienimmän luvun välillä.
4. Tutustu binääri-indeksipuuhun. Mitä hyviä ja huonoja puolia siinä on verrattuna segmenttipuuhun?
 5. Kaksiulotteinen segmenttipuu mahdollistaa kyselyt kaksiulotteisen taulukon minkä tahansa suorakulmion alueelta. Selitä, miten kaksiulotteisen segmenttipuun voi toteuttaa. Mitkä ovat aluekyselyn ja päivityksen aikavaativuudet? Piirrä seuraavaa taulukkoa vastaava segmenttipuu summien laskemiseen:

7	6	1	6
8	7	5	2
3	9	7	1
8	5	3	8

6. Lähtökohtana on lukujen $1 \dots n$ permutaatio ja tehtävänä on laskea, montako nousevaa alijonoa permutaatiossa on yhteensä. Esimerkiksi permutaatiossa $(1, 4, 2, 3)$ nousevia alijonoja on yhteensä 9: (1) , $(1, 2)$, $(1, 2, 3)$, $(1, 3)$, $(1, 4)$, (2) , $(2, 3)$, (3) ja (4) . Tee tehokas algoritmi ongelmaan ja laske sen avulla vastaus tiedostossa `permutaatio.txt` olevaan tapaukseen, jossa $n = 1000000$.

Huom.: Vastaus on suuri luku eikä mahdu 64-bittiseen muuttujaan. Jos toteutat ratkaisun Javalla, niin `BigInteger` on hyvä valinta.