



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Merkkijonohajautus

Algoritmit ongelmanratkaisussa,
kevät 2018





Tavoite

- haluamme pystyä vertaamaan tehokkaasti merkkijonojen osajonoja
- esimerkki: ovatko merkkijonon alussa olevat k merkkiä samat kuin merkkijonon lopussa olevat k merkkiä?
- helppo ratkaisu silmukalla vie aikaa $O(k)$
- hajautuksen avulla aikaa kuluu vain $O(1)$



Oheismateriaalia

- Kisakoodarin käsikirja, luku 26.2
- <https://cses.fi/kkkk.pdf>



Hajautuksen idea

- vertaamme osajonojen merkkien sijasta niiden hajautusarvoja
- hajautusarvo kertoo tiiviissä tilassa koosteen osajonon sisällöstä
- jos kaksi merkkijonoa ovat samat, niillä on sama hajautusarvo
- jos kahdella merkkijonolla on eri hajautusarvo, ne eivät ole samat



Törmäykset

- ongelma: kahdella merkkijonolla voi olla sama hajautusarvo, vaikka ne eivät ole samat
- tämä riski on aina olemassa, koska hajautusarvo voi olla paljon lyhyempi kuin sitä vastaava merkkijono
- *toivomme*, että algoritmissamme ei tapahdu törmäyksiä



Polynominen hajautus

- laskemme merkkijonon s hajautusarvon kaavalla

$$(s_0A^{n-1} + s_1A^{n-2} + \dots + s_{n-1}A^0) \bmod B$$

- tässä $s_0, s_1, \dots, s_{n-1}$ ovat merkkien koodit
- A ja B ovat etukäteen valitut vakiot



Polynominen hajautus

- esikäsittely: laskemme jokaisen merkkijonon alkuosan hajautusarvon ajassa $O(n)$
- tämän jälkeen voimme laskea minkä tahansa osajonon hajautusarvon ajassa $O(1)$ summataulukon tyyliä
- kun $h(a, b)$ on välin $[a, b]$ hajautusarvo, niin

$$h(a, b) = (h(0, b) - h(0, a - 1)A^{b-a+1}) \bmod B$$



Hajautuksen käyttö

- merkkijonohajautus on yleistyökalu, jonka avulla voimme ratkaista suuren määrän ongelmia tehokkaasti
- kaikki perustuu siihen, että voimme tarkastaa $O(1)$ -ajassa, ovatko kaksi osajonoa samat
- lisäksi voimme binäärihaun avulla selvittää kahden osajonon aakkosjärjestyksen ajassa $O(\log n)$



Hajautuksen riskit

- jos algoritmissa tulee törmäyksiä, se saattaa toimia väärin
- on tärkeää valita parametrit A ja B sopivasti ja/tai käyttää monta hajautusarvoa
- A on sivuroolissa, B kertoo suunnilleen montako eri hajautusarvoa voi tulla
- usein hyvä valinta on $B \approx 10^9$, tällöin myös tyyppi `long long` riittää laskuissa



Syntymäpäiväparadoksi

- vaikeampi tapaus on, jos vertaamme ”kaikkia hajautusarvoja keskenään”
- esimerkki: onko kahta k :n pituista osajonoa, joiden hajautusarvot ovat samat
- tällöin törmäyksiä tulee helposti vaikka olisi $B \approx 10^9$
- vastaava *syntymäpäiväparadoksi*: jos huoneessa on n henkilöä, joillakin kahdella on todennäköisesti sama syntymäpäivä vaikka n olisi varsin pieni



Syntymäpäiväparadoksi

- ratkaisu: käytetään vakiota $B \approx 10^{18}$
- voimme myös laskea kaksi eri hajautusarvoa eri parametreilla A ja B niin, että $B \approx 10^9$
- kaksi hajautusarvoa on helpompi toteuttaa, koska tyyppi `long long` riittää