

A Bittien nollaus

Havainto: On aina mahdollista muodostaa optimaalinen ratkaisu, jossa mikään alkuosan muuttaminen ei mene loppuosan muuttamisen päälle. Syy: Alkuosan ja loppuosan yhteinen osa muuttuu kaksi kertaa, joten se ei käytännössä muutu lainkaan.

Tämän havainnon jälkeen bittijonon voi jakaa kahteen osaan. Vasemmassa osassa on pelkkiä alkuosan muutoksia ja yksittäisiä muutoksia, kun taas oikeassa osassa on pelkkiä loppuosan muutoksia ja yksittäisiä muutoksia. Pienimmän muutosten määrän voi etsiä riippumattomasti vasemmalle ja oikealle puolelle. Sitten riittää käydä läpi kaikki mahdolliset jakokohdat.

B Ruudukko

Tämä on suoraviivainen laskutehtävä, jossa merkkijonon oikean kohdan saa jakojäännöksellä.

C Säde

Tehtävä jakaantuu kahteen tapaukseen:

1. säde osuu oikeassa reunassa kulmaan
2. säde ei osu kulmaan

Ensimmäisessä tapauksessa säde osuu m ruutuun, jossa m on ruudukon leveys. Pienen päättelyn jälkeen selviää, että toisessa tapauksessa säde osuu $m + n - 2$ ruutuun.

D Kielletty sana

Tehtävän voi mallintaa äärellisen automaatin avulla. Automaatti tutkii, onko merkkijonossa osajonoa, jossa on kielletty sana. Jos kielletty sana löytyy, automaatti menee hylkäävään tilaan, ja kaikki muut automaatin tilat ovat hyväksyviä. Tämän jälkeen automaattia täytyy simuloida n askelta ja laskea reittien määrät eri tiloihin. Tehokas keino tähän on käyttää dynaamista ohjelmointia.

E Murtoluvut

Tämä on suoraviivainen tehtävä, joka saattaa olla kuitenkin yllättävän vaikea saada toteutettua oikein. Tehtävässä täytyy huomata, että 32-bittinen tyyppi (esim. Javan int) ei riitä tehtävän ratkaisuun, vaan välituloksissa saattaa tarvita 64-bittistä tyyppiä (esim. Javan long).

F Labyrintti

Tehtävän voi ratkaista suorittamalla leveyshaun, joka lähtee liikkeelle yhtä aikaa kaikista ruudukon reunoista. Ruutu, johon leveyshaku päättyy viimeisenä, on kauimpana reunoista oleva ruutu.