

## A Numeropeli

Erilaisia ruudukkoita on  $9! = 362880$ , joten kaikkien tapausten vastaukset voi laskea etukäteen muistiin. Tämä onnistuu aloittamalla leveyshaulla järjestyksessä olevasta ruudukosta, jolloin löytyy pienin siirtomäärä kaikkiin muihin ruudukoihin.

Mikä tahansa ruudukko on ratkaistavissa, joten tehtävänannossa oleva -1 on hämäystä.

## B Bittijono

Tehtävän voi ratkaista dynaamisella ohjelmoinnilla laskemalla jokaiselle bittijonon alkuosalle pienin ja suurin mahdollinen bittien määrä. Joka vaiheessa haku haarautuu kahteen vaihtoehtoon: viimeiseen kohtaan tulee bitti 0 tai bitti 1. Toteutus onnistuu ajassa  $O(n)$  aputaulukoiden avulla.

## C Lukuväli

Tehokas ratkaisu onnistuu kahden osoittimen avulla, joista toinen osoittaa jäljelle jäävän lukujonon ensimmäiseen lukuun ja toinen osoittaa viimeiseen lukuun. Joka askeleella ensimmäinen osoitin liikkuu askeleen eteenpäin ja toinen osoitin liikkuu niin pitkälle eteenpäin kuin on mahdollista, ennen kuin se törmää ensimmäiseen osoittimeen. Tätä varten tarvitaan taulukko, joka kertoo jokaisesta luvusta, missä kohtaa taulukkoa se sijaitsee.

## D Laatat

Ratkaistavuuden ehdon voi ilmaista muutamalla if-lauseella, mutta ratkaisun keksiminen voi olla vaikeaa etenkin kisatilanteessa. Hyvä keino on toteuttaa ensin brute-force-ratkaisu ja tutkia sen avulla, milloin ratkaisu on mahdollista pienissä tapauksissa. Tämän jälkeen voi päätellä yleisen säännön – ja todistaa sen, jos tuntuu siltä.

## E Nollaus

Tehtävän voi ratkaista dynaamisella ohjelmoinnilla niin, että jokaisesta mahdollisesta luvusta tallennetaan muistiin, mikä on lyhin sarja nollata kyseinen luku.

## F Nimilista

Suoraviivainen ratkaisu on luoda aputaulukko, joka sisältää alkuperäisen taulukon nimet järjestyksessä. Tämän jälkeen riittää tarkistaa, moniko nimi on valmiiksi oikealla paikallaan. Jos tasan kaksi nimeä on oikealla paikalla, ratkaisu on mahdollinen.