

## BROJ

Znamenke iz ulaznog niza dijelimo na dva niza. U jedan niz stavljamo parne znamenke, a u drugi neparne. Nizove zatim sortiramo silazno.

Broj možemo konstruirati na dva načina – prva znamenka će biti ili parna ili neparna, a zatim svaka sljedeća znamenka mora biti različite parnosti. Kako bi brojevi bili što veći, znamenke stavljamo u silaznom poretku. Rješenje je veći od dva dobivena broja (jedan s parnom vodećom znamenkom, a drugi sa neparnom).

## HOTEL

Odaberimo prvi stupac koji će sadržavati premijera na sve moguće načine (ima ih točno  $L-K+1$ ). Odabir prvog stupca automatski određuje sve stupce koji sadrže premijera.

Pronađimo najniži stupac među stupcima koji sadrže premijera i neka je on visine  $h_{\min}$ . Kako ne možemo dodavati goste već samo ih izbacivati, potrebno je sve stupce koji sadrže premijera svesti na visinu  $h_{\min}$ . Iznimno, ako premijer i dalje ne stane u hotel (jer je  $h_{\min} + K > H$ ), sve stupce je potrebno svesti na visinu  $H-K$ .

Među svim mogućim pozicijama premijera odabiremo onu za koju je potrebno izbaciti najmanje gostiju.

## TABLICA

Rješenje koje redom simulira transformacije je složenosti  $O(N^2 \cdot T)$ , te je presporo za dana ograničenja. Ključna dosjetka za efikasno rješenje jest ta da je transformacija reverzibilna, tj. da svaku transformaciju možemo poništiti primjenom iste transformacije rotirajući u suprotnom smjeru.

Kako znamo konačni položaj broja koji tražimo, možemo obrnutom transformacijom naći gdje se nalazio u prethodnom koraku. Koristeći istu logiku redom za sve transformacije, primjenom obrnutih transformacija u obrnutom poretku (od zadnje prema prvoj) možemo odrediti početni položaj broja, a time i broj.

Ušteda u odnosu na naivni algoritam je u tome što nije potrebno rotirati cijelu tablicu, već pratiti položaj samo jednog broja.