

## NOGA

Kako su ograničenja na broj skakavaca i broj skokova vrlo velika, rješenje koje na jednostavan način simulira svaki skok je presporo. Potrebno je red skakavaca ostvariti na nešto pametniji način kako ne bismo za svaki skok morali prelaziti sve preskočene skakavce, tražeći najvećeg među njima.

Jedan od načina za rješavanje ovog zadatka jest podijeliti red skakavaca na  $K$  blokova tako da svaki blok sadrži  $N/K$  skakavaca. Ako unutar svakog bloka poredak skakavaca pamtimo u vezanoj listi, a visine skakavaca u gomili (eng. heap), tada možemo svaki skok obraditi u nešto kraćem vremenu.

Jedan skok s ovako strukturiranim podacima obrađujemo na sljedeći način:

- Blok u kojem skok započinje, te blok u kojem skok završava obrađujemo jednostavnim prolaskom kroz sve preskočene skakavce. Složenost obrade ovog dijela skoka jest  $O(N/K)$ .
- Blokove koje skakavac u potpunosti preskoči obrađujemo tako da pogledamo element koji se nalazi na vrhu gomile (to je najveći skakavac u bloku), a zatim skakavac s jednog kraja bloka pošaljemo u susjedni blok, a na drugom kraju bloka primamo skakavca iz susjednog bloka sa te strane, i to ovisno o tome radi li se o skoku ulijevo ili udesno. Složenost obrade ovog dijela skoka jest  $O(K \log N/K)$ .

Kako je ukupna složenost obrade skoka jednaka  $O(N/K + K \log N/K)$ , odabiremo  $K$  takav da dana funkcija ima minimum za taj  $K$ . U službenim rješenjima je za  $K$  odabrana vrijednost 100.

## POVEZ

Zadatak možemo modelirati automatom s konačnim brojem stanja, gdje svako stanje predstavlja jednu šalicu te iz svakog stanja postoje točno dva prijelaza, po jedan za svako slovo. Na početku ne znamo u kojem se automat nalazi pa kažemo da su sva stanja moguća. Potrebno je naći ulaz koji će automat dovesti u prvo stanje, bez obzira u kojem se stanju nalazio na početku, tj. naći će ulaz nakon kojeg je samo prvo stanje moguće.

Ako odaberemo par stanja ( $A$ ,  $B$ ) te nađemo niz znakova koji automat iz stanja  $A$  dovodi u stanje 1, te iz stanja  $B$  u stanje 1, tad tim ulazom sigurno smanjujemo broj mogućih stanja. Ovo možemo uvijek napraviti jer, ako postoji ukupno rješenje (što je garantirano u zadatku), tad postoji i niz znakova koji bilo koja dva stanja dovodi u stanje 1.

Ideja rješenja je održavati skup mogućih stanja, odabrati neka dva stanja, naći niz znakova koji stopi ta dva stanja, te izračunati novi skup mogućih stanja (koji je sigurno manji od prethodnog). Nakon nekog broja iteracija ostat će samo jedno moguće stanje, koje je lako dovesti u prvo.

Službeno rješenje uvijek odabire prva dva moguća stanja – jedno od njih je uvijek stanje 1.

Niz znakova koji dovodi neka dva stanja u prvo možemo naći pretraživanjem u širinu, a možemo i jednim pretraživanjem u širinu na obrnutom grafu naći takav niz za sve parove stanja odjednom.