

Zadatak TRAMPOLIN – opis algoritma.

S obzirom na Kickassov početni položaj razlikujemo dva slučaja:

1. slučaj (pojavljuje se u 20% test podataka): nije moguće stići do trampolina.
Moguće je pomisliti da je ovdje dovoljno usporediti broj posjećenih nebodera ako se krećemo samo na lijevo (dok god možemo), te samo na desno. Postoje međutim još dvije mogućnosti: kretanje na lijevo po neboderima iste visine (ako takvi postoje) i tek potom kretanje na desno dok god možemo, te obrnutno. Provjeriti ove mogućnosti te izabrati bolju vrlo je jednostavno.
2. slučaj (pojavljuje se u 80% test podataka): moguće je stići do trampolina.
Pretpostavimo da su A, B, C, ..., M oznake svih nebodera na kojima se nalazi trampoline ili s kojih je moguće stići do nekog trampolina (među njima se nalazi i neboder K) – nazovimo te nebodere lijepima. Neka su T(A), T(B), T(C), ..., T(M) odgovarajući trampolini (među njima možda ima istih). Promotrimo sljedeći put:
$$K \rightarrow \dots \rightarrow T(K) \rightarrow A \rightarrow \dots \rightarrow T(A) \rightarrow B \rightarrow \dots \rightarrow T(B) \rightarrow C \rightarrow \dots \rightarrow T(C) \rightarrow \dots \rightarrow M \rightarrow \dots \rightarrow T(M).$$
On će dakle posjetiti sve lijepe nebodere. Potom je potrebno posjetiti najdulji mogući niz ne-lijepih nebodera (s kojih nije moguće stići do trampolina). Konačni je rezultat, dakle, broj lijepih nebodera uvećan za duljinu tog niza.
Ovo rješenje moguće je realizirati u nekoliko koraka.
 - 1) Najprije označimo lijepima sve nebodere na kojima se nalazi trampoline.
 - 2) Prolazimo po neboderima s lijeva na desno i označavamo da je i-ti neboder lijep ako je (i – 1). neboder bio lijep te možemo skočiti na nj.
 - 3) Prolazimo po neboderima s desna na lijevo i označavamo da je i-ti neboder lijep ako je (i + 1). neboder bio lijep te možemo skočiti na nj.
 - 4) Za sve nebodere koji nisu označeni lijepima ni u jednom od prethodnih koraka, računamo koliko je nebodera moguće posjetiti krenuvši od tog nebodera i idući na lijevo. To činimo dinamički, na sličan način kao u koraku 2). Analogno, duljine puteva na desno računamo slično kao u koraku 3).
 - 5) Duljina konačnog niza ne-lijepih nebodera zapravo je najveći od brojeva dobivenih u koraku 4).

Složenost algoritma u oba je slučaja $O(N)$.