

D. Sofijina osveta

Ime zadatka	Sofijina osveta
Vremensko ograničenje	4 sekunde
Memorijsko ograničenje	1 gigabajt

Poznato je da Sofija najviše voli zadatke koje predlaže Paja (i Puzić). Da bi uzvratila uslugu, ona je odlučila da zada jedan zadatak njima.

U starom delu Lunda, postoji ulica sa N kuća, poređanih u jedan red, indeksiranih od 0 do $N - 1$. Sofija je kupila jednu od ovih kuća i zatražila od Paje i Puzića da pogode koja je to kuća. Ovo pogađanje će se održati kroz igru. Pre nego što igra počne, Paja i Puzić samo znaju redne brojeve kuća u ovoj ulici. Tada, oni mogu da izaberu pozitivan ceo broj K i da se slože o njihovoj zajedničkoj strategiji. Od tog trenutka nadalje, oni više neće pričati tokom trajanja igre.

Sama igra se sastoji od dve faze. U toku prve faze, Sofija bira redosled u kojem će pokazati ove kuće Paji, ali tako da poslednja kuća koju mu pokaže bude ona koju je kupila. Paja ne zna ovaj redosled unapred. Ona mu zatim pokazuje kuće u ovom redosledu. Na svakoj kući osim poslednje, Paja ima pravo da napiše na vratima jedan ceo broj između 1 i K . Na vratima poslednje kuće, one koje pripada Sofiji, Sofija ima pravo da napiše jedan ceo broj između 1 i K .

U drugoj fazi igre, Puzić će prošetati celom ulicom i pročitati sve brojeve koje su na kućama pisali Paja i Sofija. On sada želi da pogodi koju je kuću Sofija kupila. Sofija je odlučila da mu da svega dve prilike da ispravno pogodi kuću. Ukoliko uspe, on i Paja pobeđuju u igri. U suprotnom, Sofija pobeđuje u igri.

Da li možeš da osmisliš strategiju koja će pomoći Paji i Puziću da pobeđe u ovoj igri? Strategija koju osmisliš će biti bodovana na osnovu vrednosti K (što manja vrednost, to bolje).

Implementacija

Ovo je problem u kojem će se tvoj program pokretati više puta po test primeru. Prvi put, treba da implementira Pajinu strategiju. Drugi put, treba da implementira Puzićevu strategiju.

Prva linija standardnog ulaza sadrži dva cela broja P i N , gde je P jednako 1 ili 2 (prva ili druga faza), i N je broj kuća.

Osim u ulazu primera iz teksta zadatka (koji se ne računa za bodovanje), N će uvek biti jednako 100 000.

Sledeći opis ulaza zavisi od faze o kojoj je reč

Faza 1

Tvoj program treba da počne ispisujući broj K u jednoj liniji ($1 \leq K \leq 1\,000\,000$). Zatim, $N - 1$ puta, on treba da pročita jedan broj koji predstavlja indeks i ($0 \leq i < N$), i da ispiše liniju sa brojem A_i ($1 \leq A_i \leq K$). Svaki indeks i (osim indeksa Sofijine kuće) će se pojaviti tačno jednom, u nekom redosledu u kojem je odlučio grejder.

Faza 2

Tvoj program treba da pročita jednu liniju sa N celih brojeva, A_0, A_1, \dots, A_{N-1} .

Zatim, treba da ispiše jednu liniju sa dva cela broja s_1 i s_2 ($0 \leq s_i < N$), indekse koje pogađa Puzić. Dozvoljeno je da s_1 i s_2 budu jednaki.

Posle svake linije koju ispišeš, ne zaboravi da uradiš flush standardnog izlaza, ili tvoj program može dobiti ocenu Time Limit Exceeded. U Pajtonu, `print()` radi flush automatski. U C++, `cout << endl;` takođe radi flush, pored štampanja nove linije; ukoliko koristiš `printf`, koristi `fflush(stdout)`.

Grejder za ovaj problem može biti adaptivan, što znači da može menjati svoje ponašanje na osnovu tvog izlaza, da bi sprečio razna heuristična rešenja. Možda će izvršiti fazu 1, pogledati tvoj izlaz i zatim izvršiti fazu 1 ponovo, na osnovu informacija koje je zaključio iz prvog izvršavanja.

Tvoj program mora biti deterministički, što znači da mora da se ponaša isto ako ga dva puta izvršavamo na istom ulazu. Ako želiš da koristiš random u tvom programu, moraš koristiti isti random seed. Ovo može da se uradi pozivanjem `srand` sa konstantnim argumentom (u C++-u) ili korišćenjem `random.seed` (u Pajtonu). Ne smeš da koristiš `srand (time(NULL))` u C++-u. Ako grejder detektuje da tvoj program nije deterministički, dobićeš ocenu Wrong Answer.

Ako je *suma* vremena izvršavanja (izvršavanja će biti najviše 3) tvog programa preko vremenskog ograničenja, tvoje rešenje će dobiti ocenu Time Limit Exceeded.

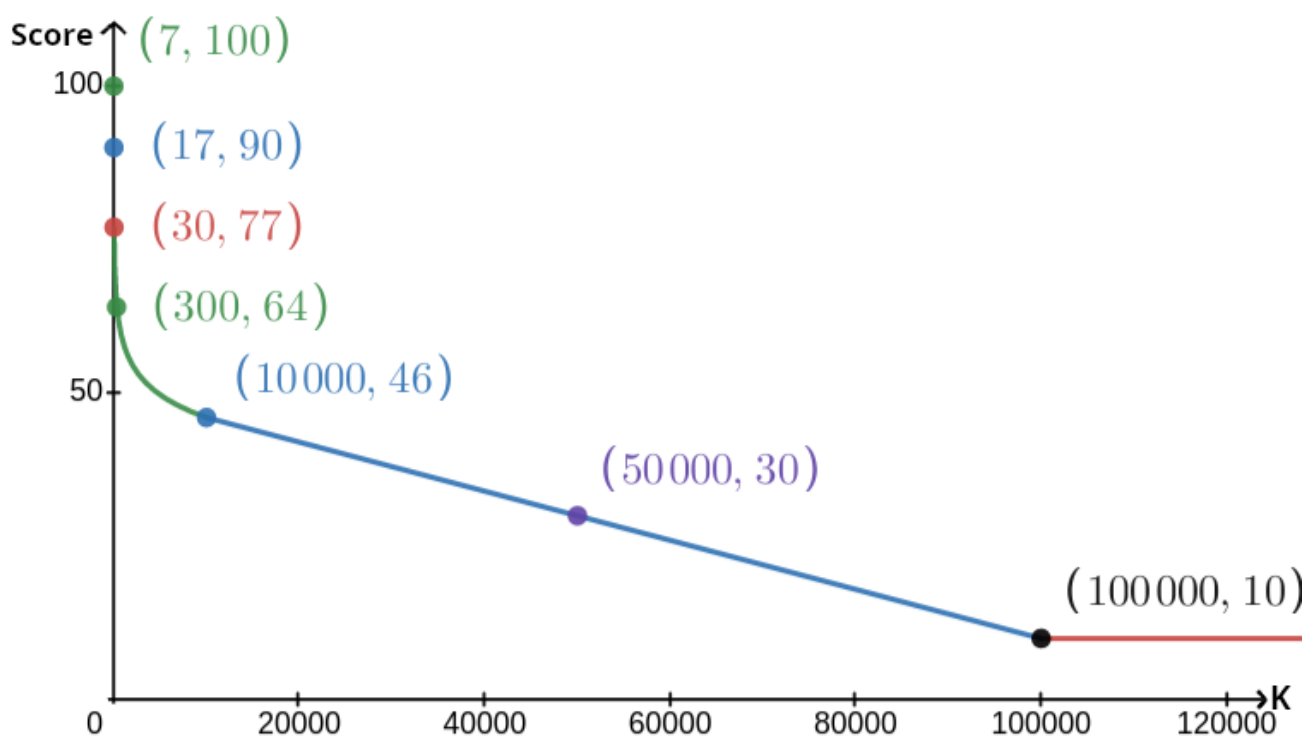
Bodovanje

Tvoje rešenje će biti testirano na više test primera. Ako na *bilo kom* od ovih test primera ne završi izvršavanje uspešno (npr. davanjem pogrešnih odgovora (Wrong Answer), crash-ovanjem (Run-Time Error), prekoračavanjem vremenskog ograničenja (Time Limit Exceeded), itd.), dobićeš 0 poena i odgovarajuću ocenu.

Ako tvoj program uspešno pronađe indeks Sofijine kuće u *svim* test primerima, dobijaš ocenu Accepted, i rezultat koji se računa na sledeći način. Neka je K najveće K koje je vaše rešenje koristilo u test primerima. U zavisnosti od K :

	Bodovi
$K > 99\,998$	10 bodova
$10\,000 < K \leq 99\,998$	$10 + \lfloor 40(1 - K/10^5) \rfloor$ bodova
$30 < K \leq 10\,000$	$46 + \lfloor 31(4 - \log_{10}(K))/(4 - \log_{10}(30)) \rfloor$ bodova
$7 < K \leq 30$	$107 - K$ bodova
$K \leq 7$	100 bodova

Funkcija za skorovanje je prikazana na slici ispod.



Test primer iz teksta zadatka se ignoriše pri bodovanju i tvoje rešenje ne mora da radi na njemu.

Testiranje

Da bi ti pomogli da testiraš svoje rešenje, prilažemo jednostavanu alatku koju možeš preuzeti. Naći ćeš je na Kattis-u pod "attachments" pri dnu stranice problema. Korišćenje ove alatke je opciono i dozvoljeno ti je da je menjaš. Primeti da zvanični grejder program na Kattis-u nije identičan ovoj alatki.

Primer korišćenja (za $N = 4$, $s = 2$, gde je s broj koji je napisan na poslednjoj kući):

Za Pajton programe, pokreni `solution.py` (obično se pokreće sa `pypy3 solution.py`):

```
python3 testing_tool.py pypy3 solution.py <<<"4 2"
```

Za C++ programe, prvo kompajliraj sa (npr. sa `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`). I zatim pokreni sa:

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

Alatka za testiranje će kuće obilaziti kuće u nasumičnom redosledu. Da bi koristili specifični red, modifikuj alatku za testiranje tamo gde piše "MODIFY HERE".

Primer interakcije

Test primer iz teksta zadatka je ignorisan pri bodovanju i tvoje rešenje ne mora da radi na njemu.

Zamisli da je $N = 4$ i da je Sofija kupila kuću sa rednim brojem 1. Neka je A lista brojeva napisanih na kućama. Na početku, $A = [0, 0, 0, 0]$, gde 0 znači da na vratima odgovarajuće kuće nije napisan nijedan broj.

Tokom prvog izvršavanja tvog programa:

$N = 4$ je dato. Tvoje rešenje odgovara sa $K = 3$.

Traži se vrednost A_2 . Tvoje rešenje odgovara sa 3. A je sada $[0, 0, 3, 0]$.

Traži se vrednost A_0 . Tvoje rešenje odgovara sa 1. A je sada $[1, 0, 3, 0]$.

Traži se vrednost A_3 . Tvoje rešenje odgovara sa 2. A je sada $[1, 0, 3, 2]$.

Konačno, grejder postavlja $A_1 = 2$, tako da je $A = [1, 2, 3, 2]$ na kraju. Ovo označava kraj prve faze.

Tokom drugog izvršavanja tvog programa, tvom rešenju je prosleđen niz 1 2 3 2.

Tvoj program odgovara sa 1 3.

Pošto je jedan od indeksa tačan (1), Paja i Puzić pobeđuju u igri.

izlaz grejdera	tvoj izlaz
1 4	
	3
2	
	3
0	
	1
3	
	2

izlaz grejdera	tvoj izlaz
2 4	
1 2 3 2	
	1 3