

D. Arvauspeli

Tehtävän nimi	Arvauspeli
Aikaraja	4 sekuntia
Muistiraja	1 gigatavu

Lundin vanhassakaupungissa on katu, jolla on N taloa rivissä numeroituna 0:sta $N - 1$:een. Emma asuu yhdessä näistä taloista ja hänen ystävänsä Aino ja Minea haluavat selvittää missä. Sen sijaan että hän kertoisi ystävilleen suoraan missä hän asuu, Emma päättää pelata pelin heidän kanssaan. Ennen pelin alkua Aino ja Minea tietävät vain kuinka monta taloa kadulla on. Tässä vaiheessa Aino ja Minea valitsevat kokonaisluvun K ja sopivat yhteisen strategian. Tämän jälkeen he eivät saa kommunikoida.

Varsinainen peli koostuu kahdesta vaiheesta. Ensimmäisessä vaiheessa Emma valitsee järjestyksen käydä talot läpi siten, että hänen omalla talollaan käydään viimeiseksi. Sitten hän vie Ainon talojen luokse tässä järjestyksessä yksi kerrallaan, kertomatta järjestystä Ainolle etukäteen. Jokaisen muun kuin Emman talon kohdalla, Aino saa kirjoittaa liidulla talon ulko-oveen yhden kokonaisluvun väliltä 1:tä K :hon. Viimeisen, eli Emman talon kohdalla, Emma kirjoittaa itse talon oveen kokonaisluvun väliltä 1:tä ja K :hon.

Pelin toisessa vaiheessa Minea kävelee katua pitkin ja lukee kaikki Ainon ja Emman oviin kirjoittamat luvut. Seuraavaksi hän haluaa arvata missä talossa Emma asuu. Hänellä on kaksi mahdollisuutta arvata oikein, ja hän voittaa Ainon kanssa, jos hän onnistuu. Muussa tapauksessa Emma voittaa pelin.

Voitko suunnitella strategian, jolla Aino ja Minea voittavat pelin varmasti? Strategiasi pisteytetään luvun K perusteella (pienempi arvo on parempi).

Toteutus

Tämä on monivaiheinen tehtävä, eli ohjelmasi suoritetaan monta kertaa. Ensimmäisellä suorituskerralla se toteuttaa Ainon strategian, ja toisella suorituskerralla Minean strategian.

Syötteen ensimmäisellä rivillä on kaksi kokonaislukua P ja N , joista P on joko 1 tai 2 (ensimmäinen tai toinen vaihe) ja N on talojen määrä. **Esimerkkisyötettä lukuun ottamatta (ei käytetä pisteytykseen), luku N on aina yhtä suuri kuin 100 000.**

Suorituksen loppuosa riippuu vaiheesta:

Vaihe 1

Ohjelmasi tulee aloittaa tulostamalla luku K yhdelle riville ($1 \leq K \leq 1\,000\,000$). Tämän jälkeen seuraava toistuu $N - 1$ kertaa: ohjelmasi tulee lukea indeksi i ($0 \leq i < N$) ja tulostaa yhdelle riville luku A_i ($1 \leq A_i \leq K$), jossa A_i on luku, jonka Aino kirjoittaa talon i oveen. Jokainen indeksi i , lukuun ottamatta Emman talon indeksia, esiintyy syötteessä täsmälleen kerran, jossakin arviointiohjelman päättämässä järjestyksessä.

Vaihe 2

Ohjelmasi tulee aloittaa lukemalla rivi, jossa on N kokonaislukua A_0, A_1, \dots, A_{N-1} .

Tämän jälkeen sen tulee tulostaa yhdelle riville kaksi kokonaislukua s_1 ja s_2 ($0 \leq s_i < N$): arvatut indeksit. Luvut s_1 ja s_2 voivat olla yhtä suuret.

Toteutuksen yksityiskohtia

Huomioi, että vaiheen 2 alkaessa ohjelmasi käynnistetään uudelleen. Tämä tarkoittaa, että et voi tallentaa tietoa muuttujiin ajojen välissä.

Varmista, että tulostepuskuri tyhjennetään jokaisen kysymyksen jälkeen, ettei ohjelmasi tulkita ylittävän aikarajaa (*Time Limit Exceeded*). Pythonissa `print()` tyhjentää puskurin automaattisesti. C++:ssa `cout << endl;` tyhjentää puskurin tulostamisen lisäksi. Jos käytät `printf`:ää, käytä lisäksi komentoa `fflush(stdout)`.

Tämän tehtävän arviointiohjelma saattaa olla mukautuva, eli se voi muuttaa käyttäytymistään riippuen ohjelmasi tulosteesta. Tämän tarkoituksena on välttää heurististen ratkaisujen hyväksyminen. Arviointiohjelma voi esimerkiksi suorittaa vaiheen 1, tarkastella ohjelmasi tulostetta ja sitten suorittaa vaiheen 1 uudestaan hyödyntäen tietoa ensimmäiseltä suorituskerralta.

Ohjelmasi pitää olla deterministinen, eli sen tulee käyttäytyä samalla tavalla, jos se suoritetaan kahdesti samalla syötteellä. Jos haluat käyttää ohjelmassasi satunnaisuutta, varmista, että käyttämäsi siemenluku on vakio. Tämän voi tehdä antamalla kovakoodattu vakio `srand`- (C++:ssa) tai `random.seed`-funktioille (Pythonissa). Et esimerkiksi saa käyttää kutsua `srand(time(NULL))` C++:ssa. Jos arviointiohjelma havaitsee, että ohjelmasi ei ole deterministinen, saat palautteen "Wrong Answer".

Jos suoritusajojen summa eri suorituskerroilta (korkeintaan kolmelta) ylittää aikarajan, ohjelmasi saa palautteen "Time Limit Exceeded".

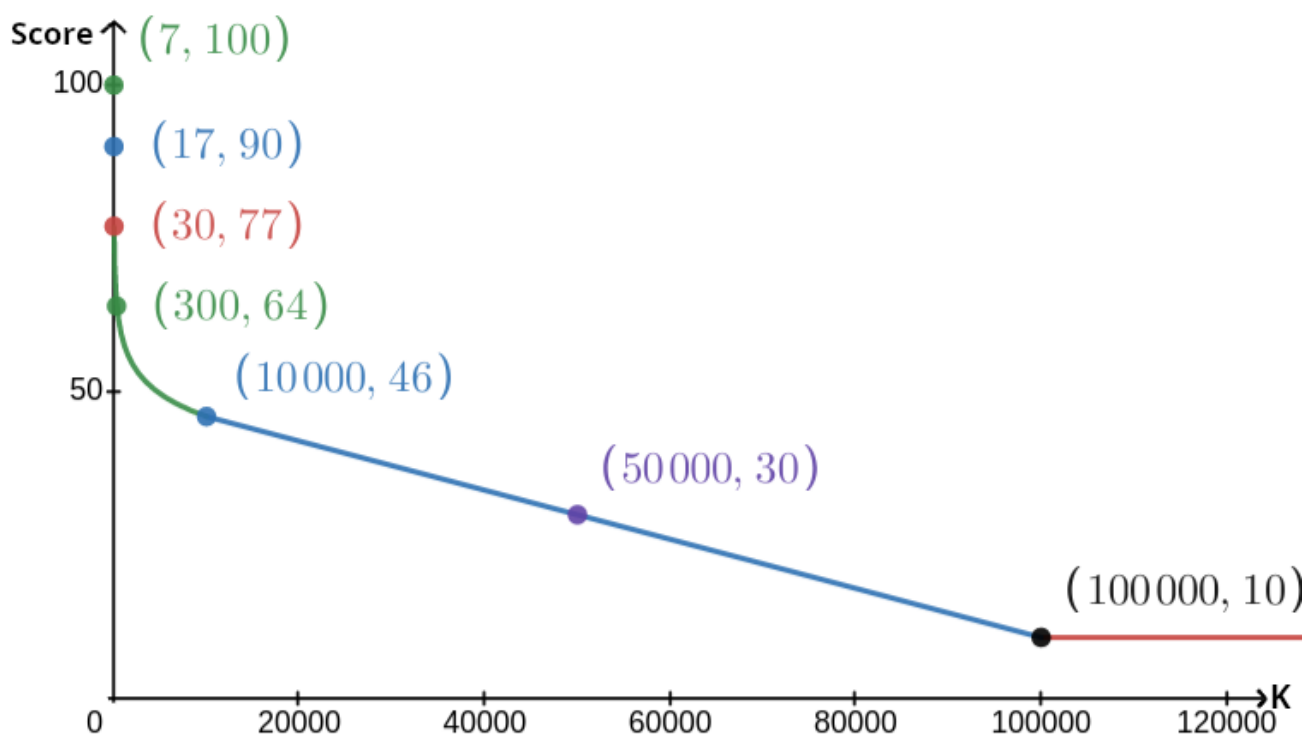
Pisteytys

Ratkaisusi testataan usealla testitapauksella. Jos ratkaisusi toimii väärin *yhdessäkään* tapauksessa (esim. tulostaa kaksi väärää arvausta (Wrong Answer), kaatuu (Runtime Error), ylittää aikarajan (Time Limit Exceeded), jne.), saat 0 pistettä ja vastaavan palautteen.

Jos ohjelmasi onnistuneesti löytää salaisen indeksin *kaikissa* testitapauksissa, saat palautteen Accepted, ja pistemääräsi lasketaan seuraavasti. Luku K_{max} tarkoittaa tässä K :n suurinta arvoa kaikissa testitapauksissa. Riippuen luvusta K_{max} :

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

Pisteytysfunktio on esitetty kuvassa alla.



Esimerkkitapausta ei oteta huomioon pisteytyksessä, eikä ratkaisusi tarvitse toimia siinä oikein.

Testaustyökalu

Ratkaisusi testaamiseksi on käytössäsi yksinkertainen ohjelma, jonka voit ladata. Ohjelma löytyy Kattiksen tehtäväsivun alalaidasta liitteistä kohdasta "attachments". Testausohjelmaa ei ole pakko

käyttää ja sitä saa muuttaa. Muista, että Kattiksessa oleva virallinen arviointiohjelma on eri kuin tämä työkalu.

Käyttöesimerkki (arvoilla $N = 4$ ja $s = 2$, joista s on viimeisenä vierailtuun taloon kirjoitettava luku):

Python-ohjelmille, kuten `solution.py` (tavallisesti suoritettuna `pypy3 solution.py`):

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

Jos käytät C++:aa, käännä ensin ohjelma (esim. `g++ -std=gnu++17 solution.cpp -o solution.out`) ja sitten suorita:

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

Testaustyökalu käy taloilla satunnaisessa järjestyksessä. Käyttääksesi tiettyä järjestystä, muuta ohjelmaa kohdasta, jossa lukee "MODIFY HERE".

Esimerkki

Esimerkkitapausta ei oteta huomioon pisteytyksessä, eikä ratkaisusi tarvitse toimia siinä oikein.

Sanotaan, että $N = 4$ ja Emma asuu talossa 1. Olkoon A lista luvuista, jotka kirjoitettiin taloihin. Aluksi $A = [0, 0, 0, 0]$. Tässä 0 tarkoittaa, että kyseiseen taloon ei ole kirjoitettu lukua.

Ohjelmasi ensimmäisellä suorituskerralla:

Annetaan luku $N = 4$. Ratkaisusi vastaa ilmoittamalla $K = 3$.

Kysytään lukua A_2 . Ratkaisusi vastaa 3. A on nyt $[0, 0, 3, 0]$.

Kysytään lukua A_0 . Ratkaisusi vastaa 1. A on nyt $[1, 0, 3, 0]$.

Kysytään lukua A_3 . Ratkaisusi vastaa 2. A on nyt $[1, 0, 3, 2]$.

Lopulta testausohjelma asettaa $A_1 = 2$, joten lopuksi $A = [1, 2, 3, 2]$. Ensimmäinen vaihe loppuu tähän.

Ohjelmasi toisella suorituskerralla ratkaisullesi annetaan lista 1 2 3 2.

Se vastaa 1 3.

Koska toinen arvauksista on oikea talon indeksi (1), Aino ja Minea voittavat pelin.

arvostelijan tuloste	ohjelmasi tuloste
1 4	
	3
2	
	3
0	
	1
3	
	2

arvostelijan tuloste	ohjelmasi tuloste
2 4	
1 2 3 2	
	1 3