

## D. Arvamismäng

Ülesande nimi	Guessing Game
Ajapiirang	4 sekundit
Mälupiirang	1 gigabait

Tartu vanalinnas on tänav, millel on  $N$  ühes reas asuvat maja, millel on indeksid 0 kuni  $N - 1$ . Emma elab ühes neist majadest ning tema sõbrad Anna ja Bertil tahavad teada, millises. Selle asemel, et lihtsalt oma sõpradele öelda, kus ta elab, otsustab Emma nendega mängu mängida. Enne mängu algust teavad Anna ja Bertil ainult majade arvu tänaval. Sel hetkel valivad nad positiivse täisarvu  $K$  ning lepivad kokku strateegia. Peale seda on igasugune omavaheline suhtlus keelatud.

Mängus endas on kaks faasi. Esimeses faasis valib Emma järjekorra, milles maju külastada, nii, et tema enda maja on neist viimane. Seejärel viib ta Anna ükshaaval selles järjekorras majade juurde, seejuures Annale eelnevalt järjekorda ütlemata. Iga maja puhul, mis pole Emma maja, on Annal lubatud selle maja esiuksle kriidiga kirjutada üks täisarv 1 ja  $K$  vahel. Viimase, Emma maja puhul kirjutab Emma ise uksele täisarvu 1 ja  $K$  vahel.

Teises mängu faasis jalutab Bertil mööda tänavat ning loeb kõiki ustele kirjutatud täisarve. Seejärel tahab ta arvata, millises majas Emma elab. Tal on õigesti arvamiseks kaks võimalust. Kui tal õnnestub õigesti arvata, võidavad mängu tema ja Anna, vastasel juhul võidab mängu Emma.

Kas suudad koostada strateegia, millega Anna ja Bertil mängu kindlasti võidavad? Sinu strateegia saab punkte  $K$  väärtuse põhjal (mida väiksem, seda parem).

## Realisatsioon

See on mitmekordse käivitamisega ülesanne ehk sinu programmi käivitatakse mitu korda. Esimesel käivitamiskorral käitab see Anna strateegiat. Teisel korral käitab see Berti strateegiat.

Esimesel sisendi real on kaks täisarvu  $P$  ja  $N$ , kus  $P$  on kas 1 või 2 (esimene või teine faas) ning  $N$  on majade arv. **Kõigis testides peale näidistesti (mida ei kasutata punktide määramisel) on  $N$  alati 100 000.**

Sellele järgnev sisend sõltub faasist.

## Faas 1

Sinu programm peaks algul väljastama täisarvu  $K$  ühel real ( $1 \leq K \leq 1\,000\,000$ ). Seejärel peaks see lugema  $N - 1$  korda rea, millel on indeks  $i$  ( $0 \leq i < N$ ), ning väljastama rea täisarvuga  $A_i$  ( $1 \leq A_i \leq K$ ), kus  $A_i$  on arv, mille Anna kirjutab maja  $i$  uksele. Iga indeks  $i$  peale Emma maja indeksi sisaldub sisendis täpselt ühe korra, hindaja määratud järjekorras.

## Faas 2

Sinu programm peaks lugema rea  $N$  täisarvuga  $A_0, A_1, \dots, A_{N-1}$ , kus  $A_i$  on maja  $i$  uksele kirjutatud arv.

Seejärel peaks see väljastama rea kahe täisarvuga  $s_1$  ja  $s_2$  ( $0 \leq s_i < N$ ), arvatavad indeksid. Pakkumised  $s_1$  ja  $s_2$  võivad olla võrdsed.

## Realisatsioonidetailid

Pane tähele, et sinu programm taaskäivitatakse enne teise faasi käitamist. See tähendab, et programm ei saa käituskordade vahel informatsiooni muutujates salvestada.

Veenduge, et programm tühjendaks väljundpuhvri peale iga rea väljastamist, vastasel juhul võib programm saada hinnangu "Ajalmiit ületatud" (*Time Limit Exceeded*). Pythonit kasutades tühjendab `print()` väljundpuhvri automaatselt. C++ kasutades väljastab `cout << endl;` reavahetuse ning tühjendab seejärel väljundpuhvri; kui kasutad `printf`, kasuta `fflush(stdout)`.

Selle ülesande hindaja **võib oma käitumist vastavalt sinu lahenduse väljunditele muuta**, eesmärgiga takistada heuristilisi (tõenäosuspõhiseid) lahendusi. Hindaja võib käitada faasi 1, vaadata väljundit ning seejärel käitada esimesest käituskorrast saadud infot kasutades uuesti faasi 1.

**Sinu programm peab olema deterministlik** ehk sama sisendi puhul samamoodi käituma. Kui soovid enda programmis juhuslikkust kasutada, siis kasuta üht ja sama juhuarvu seemneväärtust (*random seed*). Seda saab saavutada näiteks andes konstandi `srand` (C++) või `random.seed` (Python) funktsiooni parameetriks. Kui hindaja määrab, et sinu programm pole deterministlik, saab see hinnangu "Vale vastus" (*Wrong Answer*).

Kui programmi eri käivituskordade (kuni 3) käivitusaeegade *summa* ületab ajalimiiti, saab see esitus hinnangu "Ajalmiit ületatud" (*Time Limit Exceeded*).

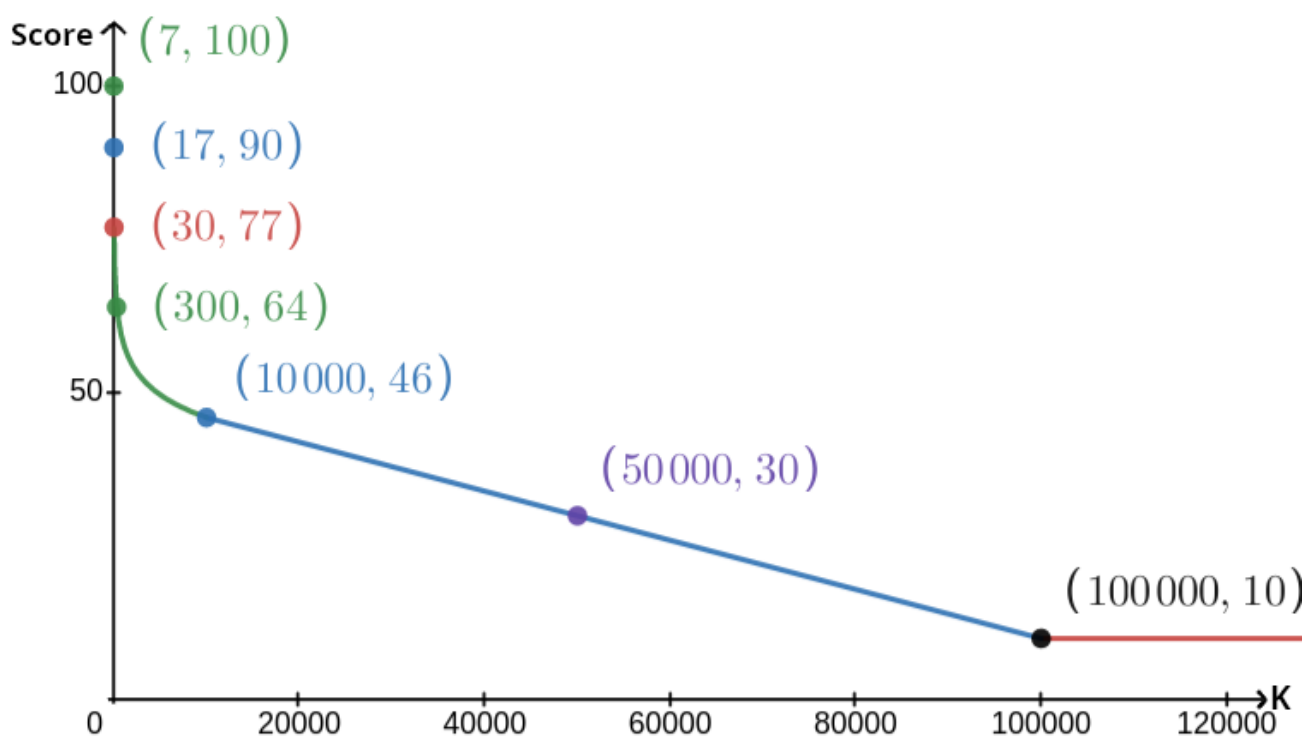
## Hindamine

Sinu lahendust testitakse hulga testidega. Kui sinu lahendus ei läbi *ükskõik missugust* neist testidest (näiteks andes valesid vastuseid (*Wrong Answer*), kokku joostes (*Runtime Error*), ületades ajalimiiti (*Time Limit Exceeded*) vms), saab lahendus 0 punkti ning vastava hinnangu.

Kui sinu programm leiab *kõigis* testides edukalt Emma maja indeksi, saab esitus hinnangu "Korras" ("Accepted") ning punktisumma, mis arvutatakse järgnevalt: olgu  $K_{max}$  maksimaalne  $K$  väärtus üle kõigi testide. Olenevalt  $K_{max}$  väärtusest:

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

Punktisumma määramise funktsiooni kujutatakse alloleval joonisel.



Näidistesti eiratakse punktide määramisel ning sinu lahendus ei pea selle puhul toimima.

## Testimistöriist

Lahenduse testimiseks on sulle antud allalaadimiseks lihtne tööriist. Vaata manuste ("attachments") osa Kattise lehe alaosas. Selle tööriista kasutamine on valikuline ning seda on lubatud muuta. Pane tähele, et ametlik hindamisprogramm Kattises erineb antud testimistöriistast.

Näidiskasutus (kus  $N = 4$ ,  $s = 2$  ning  $s$  on viimase külastatud maja uksele kirjutatav arv):

Pythoni programmi, näiteks `lahendus.py` puhul (niisama käivitamiseks `pypy3 lahendus.py`):

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

C++ programmi puhul esmalt kompileeri oma programm (näiteks käivitades `g++ -g -O2 -std=gnu++17 -static lahendus.cpp -o lahendus.out`) ning seejärel käivita:

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

Testimistööriist külastab majasid suvalises järjekorras. Kindla järjekorra kasutamiseks muuda testimistööriista kohas, kus on kirjas "*MODIFY HERE*".

## Näidissuhtlus

Näidistesti eiratakse punktide määramisel ning sinu lahendus ei pea selle puhul toimima.

Olgu näiteks  $N = 4$  ning Emma elab majas 1. Olgu  $A$  majade ustele kirjutatud arvude järjend. Algselt on  $A = [0, 0, 0, 0]$ , kus 0 tähendab, et vastava maja uksele pole arvu kirjutatud.

Esimesel programmi käituskorral:

Antakse  $N = 4$ . Sinu lahendus vastab  $K = 3$ .

Küsitakse  $A_2$ . Sinu lahendus vastab 3.  $A$  on nüüd  $[0, 0, 3, 0]$ .

Küsitakse  $A_0$ . Sinu lahendus vastab 1.  $A$  on nüüd  $[1, 0, 3, 0]$ .

Küsitakse  $A_3$ . Sinu lahendus vastab 2.  $A$  on nüüd  $[1, 0, 3, 2]$ .

Lõpuks muudab hindaja  $A_1 = 2$ , seega lõpuks  $A = [1, 2, 3, 2]$ . Sellega on esimene faas lõppenud.

Teises faasis antakse sinu lahendusele järjend  $1 \ 2 \ 3 \ 2$ .

See vastab pakkumistega  $1 \ 3$ .

Kuna üks pakkumistest on õige maja indeks (1), võidavad mängu Anna ja Bertil.

hindaja väljund	sinu väljund
1 4	
	3
2	
	3
0	
	1
3	
	2

hindaja väljund	sinu väljund
2 4	
1 2 3 2	
	1 3