

D. Hádací hra

Název úlohy	Guessing Game
Časový limit	4 sekundy
Paměťový limit	1 gigabyte

Ve velmi starém středu Lundu letos stále stojí uzoučká ulice společně s N nádherně dekorovanými domy vyrovnanými v řadě. Říkali jste, jak Švédi šikovně překládali problémy, jenže je to těžký. Proto promiňte, začneme znova:

V historickém centru Lundu se nachází ulice s N domy v řadě očíslovanými od 0 do $N - 1$. Emma žije v jednom z těchto domů a její kamarádi Anna a Bertil by chtěli zjistit, ve kterém. Místo toho, aby jim to prostě řekla, si s nimi Emma chce zahrát hru. Před začátkem hry Anna a Bertil ví pouze kolik je v ulici domů. V tomto okamžiku si Anna a Bertil mohou zvolit přirozené číslo K a domluvit se na strategii. Jakákoli další komunikace po začátku hry je zakázána.

Hra samotná se skládá ze dvou fází. V první fázi se Emma rozhodne, v jakém pořadí budou domy navštíveny, přičemž její dům bude poslední. Následně Emma postupně provede Annu těmito domy ve zvoleném pořadí, aniž by Anně toto pořadí předem prozradila. Na dveře každého domu kromě posledního (který patří Emmě) Anna napíše celé číslo od 1 do K . Nakonec Emma napíše na dveře svého domu (posledního navštíveného) číslo od 1 do K .

V druhé fázi hry Bertil projde celou ulicí od domu 0 až po dům $N - 1$ a přečte si čísla na všech dvěřích. Následně je jeho cílem uhádnout, kde Emma bydlí. Má na to dva pokusy. Pokud se jedním z nich trefí, Anna a Bertil vyhráli, v opačném případě vyhraje Emma.

Dokážete vymyslet strategii, která zaručí, že Anna a Bertil vyhrají? Vaše strategie dostane body v závislosti na hodnotě K (čím menší, tím lepší).

Implementace

Tohle je multi-run úloha, což znamená, že váš program bude spuštěn více než jednou. V prvním běhu bude implementovat Anninu strategii, v druhém běhu Bertilovu.

První řádek vstupu bude obsahovat dvě celá čísla P a N , kde P je buď 1 nebo 2 (první nebo druhá fáze), a N je počet domů v ulici. **Kromě ukázkového vstupu (který není bodovaný) se N bude vždy rovnat 100 000.**

Následující vstup závisí na fázi.

Fáze 1

Váš program by měl začít vypsáním přirozeného čísla K na samostatný řádek ($1 \leq K \leq 1\,000\,000$). Potom by měl $N - 1$ krát přečíst řádek s celým číslem i ($0 \leq i < N$) a vypsát řádek s číslem A_i ($1 \leq A_i \leq K$) udávajícím, jaké číslo Anna napíše na dveře domu i . Každý index kromě indexu Emmina domu se objeví právě jednou v pořadí zvoleném graderem.

Fáze 2

Váš program by měl přečíst řádek s N celými čísly A_0, A_1, \dots, A_{N-1} , kde A_i je číslo napsané na dvěřích domu i .

Následně by měl vypsát dvě celá čísla s_1 a s_2 ($0 \leq s_i < N$), indexy, které Bertil tipuje. Indexy s_1 a s_2 se mohou rovnat.

Implementační detaily

Před spuštěním fáze 2 bude váš program restartován. Nemůžete si tedy pamatovat žádné informace ani proměnné mezi jednotlivými běhy.

Ujistěte se, že po každém výpisu flushujete standardní výstup, jinak by vaše řešení mohlo dostat verdikt Time Limit Exceeded. V Pythonu `print()` flushuje automaticky. V C++ `cout << endl;` flushuje a vypíše nový řádek. Pokud používáte `printf`, můžete flushnout výstup pomocí `fflush(stdout)`.

Grader v této úloze může být **adaptivní**, což znamená, že jeho chování může záviset na chování vašeho programu. Grader tímto může eliminovat heuristická řešení. Grader může spustit fázi 1, podívat se na váš výstup, a potom spustit fázi 1 s nově získanými informacemi.

Vaše řešení musí být deterministické. To znamená, že když se váš program spustí dvakrát na stejném vstupu, musí se v obou případech chovat stejně. Pokud chcete ve svém řešení využívat náhody, musíte použít fixní seed. To můžete udělat tak, že ve svém programu zavoláte `srand` (v C++) nebo `random.seed` (v Pythonu) s pevnou konstantou. Nemůžete tedy například zavolat `srand(time(NULL))` v C++. Pokud grader zjistí, že váš program není deterministický, dostanete verdikt Wrong Answer.

Pokud *součet* časů jednotlivých běhů (až tří) vašeho programu přesáhne časový limit, vaše řešení obdrží verdikt Time Limit Exceeded.

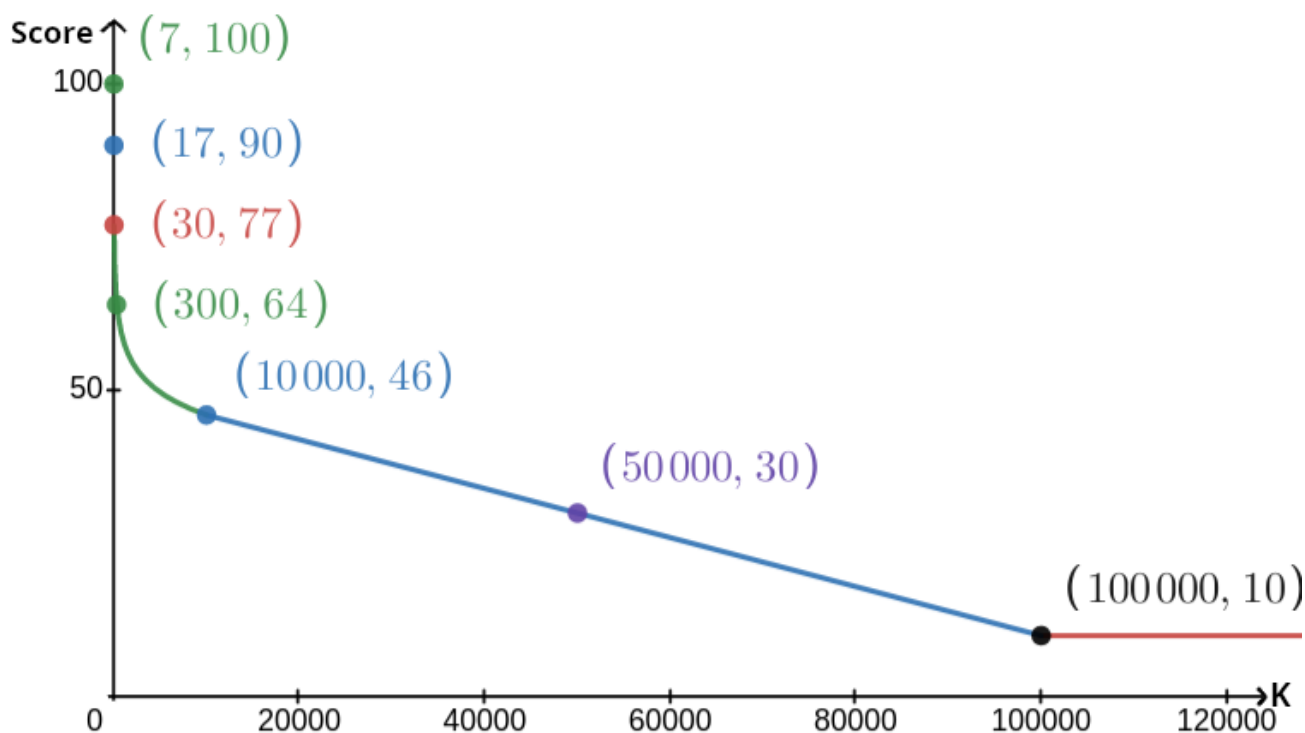
Bodování

Vaše řešení bude testováno na jistém počtu testů. Pokud vaše řešení selže na *libovolném* z těchto testů (např. odpoví špatně (Wrong Answer), spadne (Run-Time Error), překročí časový limit (Time Limit Exceeded), atd.), získáte 0 bodů a obdržíte příslušný verdikt.

Pokud váš program úšesně najde index domu, kde Emma bydlí, ve *všech* testech, získáte verdikt Accepted a body budou spočítány následujícím způsobem. Necht' K_{max} je maximální hodnota K přes všechny testy. V závislosti na K_{max} :

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

Bodovací funkce je zakreslena na obrázku níže.



Ukázkový příklad se nepočítá do hodnocení a vaše řešení na něm nemusí fungovat.

Testovací nástroj

Pro lokální testování vašeho řešení máte k dispozici jednoduchý nástroj, který si můžete stáhnout. Najdete jej v "attachments" dole na Kattis stránce této úlohy. Použití tohoto nástroje je nepovinné a

můžete jej libovolně měnit. Oficiální grader v Kattisu je jiný, než testovací nástroj.

Příklad použití (s parametry $N = 4$, $s = 2$, kde s je číslo na posledním navštíveném domě):

Programy v Pythonu, například `solution.py`, (který byste mohli spustit `pypy3 solution.py`):

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

Programy v C++, například `solution.cpp`, nejprve zkompilejte (například pomocí `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) a potom spustíte:

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

Testovací nástroj navštíví domy v náhodném pořadí. Pokud chcete zvolit jiné pořadí, opravte testovací nástroj na místě, kde se píše "MODIFY HERE".

Příklad interkace

Ukázkový příklad se nepočítá do bodování a vaše řešení na něm nemusí fungovat.

Nechť $N = 4$ a Emma bydlí v domě 1. Nechť A je seznam čísel napsaných na domech. Na začátku $A = [0, 0, 0, 0]$, kde 0 znamená, že na daném domě zatím není napsané žádné číslo.

V prvním běhu vašeho programu:

Je dáno $N = 4$. Vaše řešení odpoví $K = 3$.

Grader se zeptá na A_2 . Vaše řešení odpoví 3. A je nyní $[0, 0, 3, 0]$.

Grader se zeptá na A_0 . Vaše řešení odpoví 1. A je nyní $[1, 0, 3, 0]$.

Grader se zeptá na A_3 . Vaše řešení odpoví 2. A je nyní $[1, 0, 3, 2]$.

Grader nakonec nastaví $A_1 = 2$, takže na konci $A = [1, 2, 3, 2]$. Toto je konec první fáze.

Následuje fáze 2. Vaše řešení dostane seznam 1 2 3 2.

Vaše řešení odpoví 1 3.

Protože jeden z tipů je index správného domu (1), Anna a Bertil vyhráli.

Výstup graderu	Váš výstup
1 4	
	3
2	
	3
0	
	1
3	
	2

Výstup graderu	Váš výstup
2 4	
1 2 3 2	
	1 3