

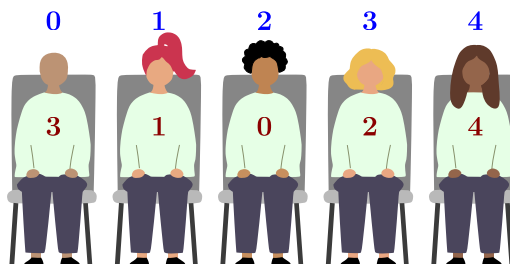
D. Sætafyrirkomulag (seatingplan)

Á lokahátíð þessarar EGOI keppni munu N mikilvægir gestir mæta. Þeir þurfa allir að sitja í fremstu röð í mjög ákveðinni röð sem er í samræmi við öll smáatriði diplómatískra samskipta. Það tók Noemi tvær andvökunætur að ákvarða rétta sætafyrirkomulagið.

Veronika hefur umsjón með lokahátíðinni. Ein af mörgum skyldum hennar er að ganga úr skugga um að sætin í fremstu röð hafi rétt nafnaskilti. Það er aðeins eitt lítið vandamál: Noemi sagði henni aldrei hver rétta sætaskipanin væri, og nú er Noemi hvergi að finna. Sem betur fer er Dorka ljósmyndari með smáforrit sem gæti komið að gagni.

Dorka þurfti að undirbúa myndavélarnar sínar svo hún gæti tekið ákveðnar myndir af gestum í fremstu röð. Fyrir uppsetninguna þurfti hún að vita hversu breið hver mynd yrði, svo Noemi bjó til smáforrit handa henni sem skilar fljótt þeim upplýsingum sem hún þarf. Veronika vill nú nota smáforritið til að finna rétta sætafyrirkomulagið.

Hinir N mikilvægu gestir eru tölusettir frá 0 til $N - 1$. Sætin í fremstu röð eru einnig tölusett frá 0 til $N - 1$, frá vinstri til hægri. Fyrir hvert I ($0 \leq I \leq N - 1$), látum g_I tákna gestinn sem á að sitja í sæti I , og látum s_I tákna sætið sem gestur I á að sitja í.



Mynd 1: Röð með fimm gestum. Fyrir þessa röð er $g = [3, 1, 0, 2, 4]$ og $s = [2, 1, 3, 0, 4]$.

smáforritið virkar sem hér segir:

- Dorka slær inn númerin I , J , K fyrir nákvæmlega þrjá mismunandi gesti.
- Smáforritið segir henni lágmarksfjölda gesta sem verða sýnilegir ef allir þrír völdu gestirnir eru á myndinni. Formlega mun smáforritið birta gildið $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$.

Til dæmis, skoðum stöðuna sem sýnd er í Mynd 1:

- Gestir $I = 0$, $J = 2$ og $K = 4$ eru í sætum $s_I = 2$, $s_J = 3$ og $s_K = 4$. Ef Dorka velur þá, mun smáforritið birta gildið $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$.

Með öðrum orðum, mjóasta myndin sem inniheldur gesti 0, 2, og 4 hefur einungis þessa þrjá gesti.

- Gestir $I = 0$, $J = 4$, og $K = 3$ eru í sætum $s_I = 2$, $s_J = 4$ og $s_K = 0$. Ef Dorka velur þá, mun smáforritið birta gildið $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$.

Með öðrum orðum verður mynd sem inniheldur þessa þrjá tilteknu gesti að innihalda alla 5 gestina.

Hjálpðu Veroniku að ákvarða réttu sætaskipanina með því að nota smáforrit Dorku. Nánar tiltekið á forritið þitt að ákvarða og skrifa út rununa g_0, g_1, \dots, g_{N-1} . Það eru alltaf nákvæmlega tvö rétt svör

(eitt er öfug röð af hinu), og þú mátt skrifa út hvort sem er. Stigagjöfin þín mun ráðast af fjölda fyrirspurna til smáforritsins sem lausnin þín gerir.

Útfærsla



Þetta er gagnvirkt verkefni. Forritið þitt mun nota staðalinntak og staðalúttak til að eiga samskipti við yfirferðarforrit á því sniði sem lýst er hér að neðan.

Forritið þitt skal byrja á því að lesa eina línu af inntaki sem inniheldur jákvæða heiltölu T , fjölda prófunartilvika sem fylgja.

Fyrir hvert prófunartilvik ætti forritið þitt að byrja á því að lesa eina línu af inntaki sem inniheldur jákvæða heiltölu N , fjölda sæta, sem er einnig fjöldi gesta.

Til að gera fyrirspurn ætti forritið þitt að skrifa út línu á sniðinu „? $I\ J\ K$ “, þar sem $0 \leq I, J, K \leq N - 1$ eru þrjár **ólíkar** tölur.

Eftir að hafa gert fyrirspurn ætti forritið þitt að lesa eina línu sem inniheldur eina jákvæða heiltölu, svarið við fyrirspurninni þinni.

Til að svara með réttri sætaskipan ætti forritið þitt að skrifa út línu á sniðinu „! $g_0 \dots g_{N-1}$ “.

Eftir að hafa leyst öll T prófunartilvikin ætti forritið þitt að hætta keyrslu á eðlilegan hátt.

Athugaðu að opinbera yfirferðarforritið sem notað er í CMS til að prófa lausnina ykkar gæti **aðlagð** sig að lausn þinni. Það er að segja, fyrir sum prófunartilvik er umröðun gesta ekki ákvörðuð fyrirfram. Þess í stað gæti yfirferðarforritið ákveðið hvaða af eftirstandandi umröðunum það notar háð þeim fyrirspurnum sem forritið þitt hefur nú þegar gert.

Sturta úr staðalúttaki. Ef þú notar ekki meðfylgjandi sniðmát skaltu ganga úr skugga um að sturta úr staðalúttaki eftir að hafa prentað hverja línu, annars gæti forritið þitt verið dæmt sem *Not correct*. Í Python gerist þetta sjálfkrafa ef þú notar `input()` til að lesa línur, og þú getur notað `print(..., flush=True)` til að þvinga forritið til að sturta. Í C++ framkvæmir `cout << endl;` tæmingu á biðminni auk þess að prenta nýja línu; ef þú notar `printf`, notaðu `fflush(stdout)`.

Takmarkanir

- $1 \leq T \leq 10$.
- N verður 5 (aðeins í sýnidæminu), 8, 40, eða 2000.
- Fyrir hvert prófunartilvik máttu gera í mesta lagi 10 000 fyrirspurnir.

Stigagjöf

Forritið þitt verður prófað á nokkrum prófunartilvikum sem flokkuð eru í hlutverkefni. Til að fá stig fyrir hlutverkefni verður þú að leysa öll prófunartilvikin sem það inniheldur rétt.

- **Hlutverkefni 0 [0 stig]:** Sýnidæmið ($N = 5$).
- **Hlutverkefni 1 [9 stig]:** $N = 8$.
- **Hlutverkefni 2 [11 stig]:** $N = 2000$, og gestir 0 og 1 sitja hlið við hlið.
- **Hlutverkefni 3 [15 stig]:** $N = 40$.
- **Hlutverkefni 4 [65 stig]:** $N = 2000$.

Fyrir hlutverkefni 1 og 2 mun sérhver lausn sem leysir öll prófunartilvik rétt fá öll stig.

Fyrir hlutverkefni 3 og 4 verður lausnin þín að leysa öll prófunartilvik rétt til að fá nokkur stig, og stigin munu ráðast af Q_s , mesta fjölda fyrirspurna sem lausnin þín þurfti að gera til að leysa prófunartilvik. Látum $X_s = \max(1, Q_s/N)$. Stigin fyrir hlutverkefni 3 og 4 eru síðan reiknuð sem hér segir:

$$\text{score}_3 = \min\left(15, 3 + \frac{19}{X_s^{1.5}}\right), \quad \text{score}_4 = \min\left(65, 3 + \frac{91}{X_s^{1.5}}\right)$$

Gildið score_s er námundað að næstu heiltölu fyrir hvert hlutverkefni, og heildarstigafjöldi þinn er summan af þeim. Til að fá fullt hús stiga þarftu að leysa hlutverkefni 3 í í mesta lagi 55 fyrirspurnum og hlutverkefni 4 í í mesta lagi 2597 fyrirspurnum. Dæmi um gildi á Q_s og stig fyrir hlutverkefni 3 og 4 eru sýnd hér að neðan.

Q_s	55	56	60	70	80	100	150	10000
score_3	15	14	13	11	10	8	6	3

Q_s	2597	2800	3000	4000	5000	6000	8000	10000
score_4	65	58	53	35	26	21	14	11

Dæmi

Grader	Lausn
1 5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

Útskýring

Sýnidæmið inniheldur eitt prófunartilvik ($T = 1$) með $N = 5$ gestum. Falda röðun gesta í þessu prófunartilviki samsvarar Mynd 1.

Fyrsta fyrirspurnin sem sýnislausnin gerir er 0, 2, 4. Svarið 3 við þessari fyrirspurn segir okkur að þessir gestir sitji, í einhverri óþekktri röð, í þremur samliggjandi sætum.

Svarið 3 við annarri fyrirspurninni segir okkur það sama um gestina 3, 0 og 1.

Við getum nú ályktað að gestur 0 hljóti að sitja í miðjunni, með gesti 2 og 4 öðru megin og gesti 1 og 3 hinu megin.

Eftir þriðju fyrirspurn getum við nú þegar verið viss um að gestirnir verði að sitja annað hvort í röðinni [3, 1, 0, 2, 4] eða í öfugri röð [4, 2, 0, 1, 3]. Við megum skrifa út hvora röðina sem er.

Kóðasniðmát og upplýsingar um yfirferð í CMS

Við mælum eindregið með því að nota meðfylgjandi kóðasniðmát fyrir C++ og Python. Þau athuga hvort samskipti við yfirferðarforritið hafi heppnast og hætta keyrslu á snyrtilegan hátt þegar svo er ekki.

Ef þú notar ekki meðfylgjandi sniðmát gæti CMS birt rangan dóm í tilvikum þar sem lausnin þín er röng. Til dæmis, í staðinn fyrir „Output isn't correct“ gætir þú fengið „Execution killed by signal“ eða „Execution timed out (wall clock limit exceeded)“.

Við mælum einnig með að nota prófunarverkfærið (sjá hér að neðan) til að prófa lausnina á eigin vél áður en hún er send inn. Prófunarverkfærið athugar úttök lausnarinnar og lætur vita ef ógildar fyrirspurnir eru notaðar.

Prófunarverkfæri

Til að auðvelda prófun á lausninni útvegum við einfalt verkfæri sem þú getur hlaðið niður frá CMS. Notkun þess er valfrjáls. Athugaðu að opinbera yfirferðarforritið á CMS er ekki það sama og prófunarverkfærið.

Til að nota verkfærið þarftu inntaksskrá. Þú getur notað meðfylgjandi sýnidæmisinntak `seatingplan.input0.txt` eða búið til þitt eigið. Inntaksskráin ætti að byrja á línu sem hefur fjölda T af prófunartilvikum og síðan ætti hún að hafa tvær línur fyrir hvert prófunartilvik: eina línu með tölunni N og síðan eina línu með tölunum g_0, g_1, \dots, g_{N-1} .

Fyrir Python forrit, til dæmis `seatingplan.py` (venjulega keyrt sem `pypy3 seatingplan.py`), er prófunarverkfærið keyrt svona:

```
python3 testing_tool.py pypy3 seatingplan.py < seatingplan.input0.txt
```

Fyrir C++ forrit, þýddu fyrst lausnina:

```
g++ -DEVAL -std=gnu++20 -O2 -pipe -static -s -o seatingplan seatingplan.cpp
```

og keyrðu svo prófunarverkfærið:

```
python3 testing_tool.py ./seatingplan < seatingplan.input0.txt
```