

D. Seating Plan (seatingplan)

ЕГОI-ийн хаалтын ёслолд N чухал зочин оролцоно.

Эдгээр зочдыг дипломат ёслолын бүх нарийн дүрэмд нийцүүлэн, урд эгнээнд маш тодорхой дарааллаар суулгах шаардлагатай. Зөв суудлын дарааллыг тогтоохын тулд Ноэми хоёр шөнийг нойргүй өнгөрүүлжээ.

Вероника хаалтын ёслолыг хариуцан зохион байгуулж байна. Түүний олон үүргийн нэг нь урд эгнээний суудлууд дээр зөв нэрийн пайз байрласан эсэхийг шалгах явдал юм.

Гэвч нэг жижигхэн асуудал гарчээ: Ноэми түүнд зөв суудлын дарааллыг огт хэлээгүй бөгөөд одоо Ноэми хаана байгаа нь мэдэгдэхгүй алга болсон байна.

Аз болоход гэрэл зурагчин Доркад хэрэг болж магадгүй нэг апп байна.

Дорка урд эгнээнд суусан зочдын тодорхой зургуудыг авахын тулд камераа бэлтгэх шаардлагатай байв. Тохиргоо хийхийн тулд зураг бүр ямар өргөнтэй гарахыг мэдэх хэрэгтэй болсон тул Ноэми түүнд хэрэгтэй мэдээллийг хурдан гаргаж өгдөг апп хийж өгчээ.

Одоо Вероника энэ аппыг ашиглан зөв суудлын хуваарийг олж мэдэхийг хүсэж байна.

N чухал зочдыг 0-ээс $N - 1$ хүртэл дугаарлана.

Урд эгнээний суудлуудыг мөн зүүнээс баруун тийш 0-ээс $N - 1$ хүртэл дугаарласан.

I бүрийн хувьд g_I нь I дугаартай суудалд суух ёстой зочныг, харин s_I нь I дугаартай зочны суух ёстой суудлыг тус тус илэрхийлнэ.

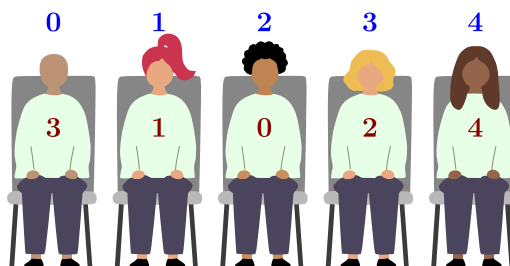


Figure 1: Таван зочинтой нэг эгнээ. Энэ эгнээний хувьд $g = [3, 1, 0, 2, 4]$ ба $s = [2, 1, 3, 0, 4]$ байна.

Апп дараах байдлаар ажиллана:

- Лиса яг гурван өөр зочны I , J , K дугааруудыг оруулна.
- Апп сонгосон гурван зочин бүгд зурагт багтахын тулд хамгийн багадаа хэдэн зочин харагдах ёстойг хэлж өгнө. Албан ёсоор, апп нь $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$ утгыг харуулна.

Жишээлбэл, Figure 1 зурагт үзүүлсэн нөхцөлийг харъя:

- $I = 0$, $J = 2$, болон $K = 4$ зочид нь $s_I = 2$, $s_J = 3$, $s_K = 4$ суудлуудад сууж байна. Хэрэв Лиса эдгээрийг сонговол, апп $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$ утгыг харуулна.

Өөрөөр хэлбэл, 0, 2, болон 4 зочдыг багтаасан хамгийн нарийн зурагт зөвхөн эдгээр гурван зочин л багтана.

- $I = 0$, $J = 4$, болон $K = 3$ зочид нь $s_I = 2$, $s_J = 4$, болон $s_K = 0$ суудлуудад сууж байна. Хэрэв Лиса эдгээрийг сонговол, апп $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$ утгыг харуулна.

Өөрөөр хэлбэл, өгөгдсөн гурван зочдыг багтаасан зурагт бүх 5 зочин багтах ёстой.

Вероникад Доркагийн аппыг ашиглан зөв суудлын дарааллыг тогтооход туслаарай.

Илүү тодруулбал, таны програм g_0, g_1, \dots, g_{N-1} дарааллыг тодорхойлж, хэвлэх ёстой.

Үргэлж яг хоёр зөв хариу байдаг бөгөөд нэг нь нөгөөгийнхөө урвуу дараалал байна. Та эдгээрийн аль нэгийг хэвлэж болно.

Таны оноо таны шийдэл апп руу хэдэн асуулга хийснээс хамаарна.

Хэрэгжүүлэлт



Энэ бол интерактив бодлого юм. Таны програм доор тайлбарласан форматын дагуу стандарт оролт болон стандарт гаралтыг ашиглан шалгагчтай харилцана.

Таны програм эхлээд дараагийн тестийн тоог илэрхийлэх эерэг бүхэл тоо T агуулсан нэг мөрийг уншина.

Тест бүрийн хувьд таны програм эхлээд суудлын тоо, мөн зочдын тоо болох эерэг бүхэл тоо N агуулсан нэг мөрийг уншина.

Асуулга хийхийн тулд таны програм “? I J K ” хэлбэрийн нэг мөрийг хэвлэх ёстой. Энд $0 \leq I, J, K \leq N - 1$ бөгөөд эдгээр нь хоорондоо ялгаатай гурван тоо байна.

Асуулга хийсний дараа таны програм тухайн асуулгын хариу болох нэг эерэг бүхэл тоо агуулсан нэг мөрийг уншина.

Зөв суудлын дарааллаар хариулахын тулд таны програм “! $g_0 \dots g_{N-1}$ ” хэлбэрийн нэг мөрийг хэвлэх ёстой.

Бүх T тестийг бодож дууссаны дараа таны програм хэвийн дуусах ёстой.

CMS дээр таны шийдлийг шалгахад ашиглах албан ёсны шалгагч нь **adaptive** буюу дасан зохицох шинжтэй байж болохыг анхаарна уу.

Өөрөөр хэлбэл, зарим тестийн хувьд зочдын permutation буюу байрлалын сэлгэмэл урьдчилан тогтоогдоогүй байж болно. Үүний оронд шалгагч нь таны програмын аль хэдийн асуусан асуулгуудаас хамааран үлдсэн сэлгэлтийн алийг нь ашиглахаа шийдэж болно.

Flushing. Хэрэв та өгөгдсөн template-үүдийг ашиглахгүй байгаа бол мөр бүрийг хэвлэсний дараа стандарт гаралтыг цэвэрлэхээ мартуузай, эс тэгвээс таны програм *Not correct* гэж үнэлэгдэж магадгүй. Python дээр хэрэв та мөрүүдийг уншихын тулд `input()` ашиглавал энэ нь автоматаар хийгддэг бөгөөд харин үүнийг цэвэрлэхийн тулд `print(..., flush=True)` ашиглаж болно. C++ хэл дээр `cout << endl`; нь шинэ мөр хэвлэхээс гадна стандарт гаралтыг цэвэрлэнэ; хэрэв `printf` ашиглаж байгаа бол `fflush(stdout)` ашиглана уу.

Хязгаарлалтууд

- $1 \leq T \leq 10$.
- N нь 5 (зөвхөн жишээнд), 8, 40, эсвэл 2000 байна.
- Тест бүрийн хувьд та хамгийн ихдээ 10 000 асуулга хийж болно.

Оноо өгөх

Таны програм хэд хэдэн тест дээр шалгагдах бөгөөд эдгээр нь дэд бодлогууд болгон бүлэглэгдсэн байна.

Дэд бодлогын оноог авахын тулд тухайн дэд бодлогод багтсан бүх тестийг зөв бодох ёстой.

- Дэд бодлого 0 [0 оноо]: Жишээ ($N = 5$).

- Дэд бодлого 1 [9 оноо]: $N = 8$.
- Дэд бодлого 2 [11 оноо]: $N = 2000$, мөн 0 ба 1 зочид хоорондоо зэрэгцээ сууна.
- Дэд бодлого 3 [15 оноо]: $N = 40$.
- Дэд бодлого 4 [65 оноо]: $N = 2000$.

Дэд бодлого 1 болон 2-ын хувьд бүх тестийг зөв бодсон ямар ч шийдэл бүтэн оноо авна.

Дэд бодлого 3 болон 4-ын хувьд оноо авахын тулд таны шийдэл бүх тестийг зөв бодох ёстой бөгөөд таны оноо Q_s -ээс хамаарна. Энд Q_s нь тухайн дэд бодлогын нэг тестийг бодоход таны шийдэлд шаардлагатай болсон асуулгын хамгийн их тоо юм.

$X_s = \max(1, Q_s/N)$ гэж тодорхойлъя.

Тэгвэл дэд бодлого 3 болон 4-ын оноог дараах байдлаар тооцно:

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

score_s -ийн утгыг дэд бодлого бүрийн хамгийн ойрын бүхэл тоо болгон тоймлодог бөгөөд таны нийт оноо нь эдгээрийн нийлбэр юм. Бүрэн оноо авахын тулд та 3-р дэд даалгаврыг хамгийн ихдээ 55 асуулгад, 4-р дэд даалгаврыг хамгийн ихдээ 2597 асуулгад бодох шаардлагатай. Q_s -ийн жишээ утгууд болон 3 ба 4-р дэд бодлогуудын оноог доор харуулав.

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

Жишээ

Грейдер	Бодолт
1	
5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

Тайлбар

Жишээ оролт нь нэг тесттэй ($T = 1$) бөгөөд $N = 5$ зочинтой. Энэ тестийн зочдын нууц байрлал нь жишээ мөртэй тохирно.

Жишээ шийдлийн эхний асуулт нь 0, 2, 4 байна. Энэ асуултын хариу 3 гэдэг нь эдгээр зочид ямар нэг үл мэдэгдэх дарааллаар хоорондоо залгаа гурван суудалд сууж байгааг илэрхийлнэ.

Хоёр дахь асуултын хариу 3 нь 3, 0, 1 зочдын хувьд мөн адил зүйлийг хэлж байна.

Иймээс зочин 0 дунд нь сууж, 2 ба 4 зочид нэг талд, 1 ба 3 зочид нөгөө талд нь сууж байгааг бид дүгнэж чадна.

Гурав дахь асуултын дараа зочид заавал [3, 1, 0, 2, 4] дарааллаар эсвэл түүний урвуу дараалал болох [4, 2, 0, 1, 3] дарааллаар суусан байх ёстой гэдэгт бүрэн итгэлтэй болно. Эдгээр хоёр дарааллын аль нэгийг хариу болгон гаргаж болно.

Code Templates and Evaluation Details in CMS

Бид C++ болон Python хэлний хувьд өгөгдсөн кодын загваруудыг ашиглахыг зөвлөж байна. Эдгээр нь үнэлэгч (Grader)-тэй харилцах харилцаа амжилттай болсон эсэхийг шалгаж, амжилтгүй болсон үед програмыг дуусгавар болно.

Хэрэв та өгөгдсөн загваруудыг ашиглахгүй бол таны шийдэл буруу тохиолдолд CMS буруу шийдвэр гаргаж магадгүй.

Жишээлбэл, “Output isn’t correct” гэсэн үгийн оронд та “Execution killed by signal эсвэл “Execution timed out (wall clock limit exceeded)” гэсэн мэдээлэл өгч магадгүй.

Мөн бид таны шийдлийг илгээхээсээ өмнө локалаар туршиж үзэхийн тулд туршилтын хэрэгслийг (доорх хэсгийг үзнэ үү) ашиглахыг зөвлөж байна. Туршилтын хэрэгсэл (Testing tool) нь таны шийдлийн гаралтыг шалгаж, мөн буруу асуулга ашигласан тухай мэдээлдэг.

Testing Tool

Шийдлээ туршихад хялбар болгохын тулд бид CMS-ээс татаж авч болох энгийн хэрэгсэл өгсөн. Энэ хэрэгслийг ашиглах эсэх нь заавал биш. CMS дээрх албан ёсны шалгагч нь энэ турших хэрэгслээс өөр гэдгийг анхаарна уу.

Энэ хэрэгслийг ашиглахын тулд оролтын файл хэрэгтэй. Та өгөгдсөн жишээ оролт болох `seatingplan.input0.txt` файлыг ашиглаж болно, эсвэл өөрөө оролтын файл үүсгэж болно.

Оролтын файл нь эхлээд тестийн тоо T -г агуулсан нэг мөрөөр эхэлнэ. Үүний дараа тест бүрийн хувьд хоёр мөр байна: нэг мөрөнд N тоо, дараагийн мөрөнд g_0, g_1, \dots, g_{N-1} тоонууд байна.

Python програмын хувьд `seatingplan.py` гэж нэрлэсэн байна гэж үзье (ихэвчлэн `py3` `seatingplan.py` гэж ажиллуулна).

Testing tool-ийг дараах байдлаар ажиллуулна:

```
python3 testing_tool.py py3 seatingplan.py < seatingplan.input0.txt
```

C++ програмын хувьд эхлээд шийдлээ compile хийнэ:

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

дараа нь турших хэрэгслийг ажиллуулна:

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