

D. Joylashtirish rejasi (seatingplan)

Ushbu EGOI yopilish marosimiga N nafar muhim mehmon tashrif buyuradi. Diplomatik protokolning nozik jihatlariga mos kelishi uchun ularning barchasi birinchi qatorda aniq tartibda o'tqizish kerak. To'g'ri joylashish tartibini aniqlash Charosning uyqusiz tunlariga sabab bo'ldi.

Habishka yopilish marosimini boshqaryapti. Uning ko'plab vazifalaridan biri birinchi qatordagi o'rindiqlarga to'g'ri ism tablichkalari qo'yilganligiga ishonch hosil qilishdir. Faqatgina bitta kichkina muammo bor: Charos unga to'g'ri joylashtirish tartibini aytmagan hamda endi Charosni hech qayerdan topib bo'lmayapti. Yaxshiyamki, fotograf Jumanazarda foydali bo'lishi mumkin bo'lgan ilova bor.

Jumanazar birinchi qatordagi mehmonlarning maxsus rasmlarini olishi uchun o'z kameralarini tayyorlashi kerak edi. Sozlash uchun u har bir rasmning kengligi qancha bo'lishini bilishi kerak bo'lgani uchun, Charos unga kerakli ma'lumotni tezda chiqarib beradigan ilova yaratib bergan edi. Endi Habishka to'g'ri joylashtirish tartibini topish uchun shu ilovadan foydalanmoqchi.

N ta muhim mehmon 0 dan $N - 1$ gacha raqamlangan. Birinchi qatordagi o'rindiqlar ham chapdan o'ngga qarab 0 dan $N - 1$ gacha raqamlangan. Har bir I ($0 \leq I \leq N - 1$) uchun g_I o'rindiq I da joylashishi kerak bo'lgan mehmonni, s_I esa I -mehmon joylashishi kerak bo'lgan o'rindiqni anglatsin.

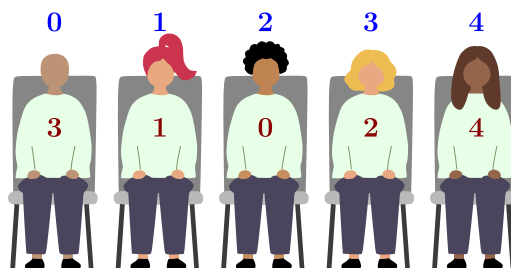


Figure 1: Besh nafar mehmonlar joylashishiga misol. $g = [3, 1, 0, 2, 4]$ va $s = [2, 1, 3, 0, 4]$.

Ilova quyidagicha ishlaydi:

- Jumanazar aynan uchta turli mehmonning I , J , K raqamlarini kiritadi.
- Ilovada, tanlangan uchala mehmon rasmda ko'rinishi uchun, rasmga eng kamida necha kishi tushishi kerakligi yozilgan. Boshqa so'zlar bilan aytganda, ilova $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$ qiymatini ko'rsatadi.

Masalan, Figure 1 da ko'rsatilgan holatni ko'raylik:

- $I = 0$, $J = 2$, va $K = 4$ mehmonlar $s_I = 2$, $s_J = 3$, va $s_K = 4$ o'rindiqlarda joylashishsin. Agar Jumanazar ularni tanlasa, ilova $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$ qiymatini ko'rsatadi.

Boshqacha qilib aytganda, 0, 2, va 4-mehmonlarni o'z ichiga olgan eng tor (eni eng kichik) rasmda faqatgina shu uchta mehmon bo'ladi.

- $I = 0$, $J = 4$, va $K = 3$ mehmonlar $s_I = 2$, $s_J = 4$, va $s_K = 0$ o'rindiqlarda joylashishsin. Agar Jumanazar ularni tanlasa, ilova $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$ qiymatini ko'rsatadi.

Boshqacha qilib aytganda, berilgan uchta mehmonni o'z ichiga olgan rasmda barcha 5 nafar mehmon ham tushishi shart.

Habishkaga Charosning ilovasidan foydalanib to'g'ri o'tirish tartibini topishga yordam bering. Aniqroq qilib aytganda, dasturingiz g_0, g_1, \dots, g_{N-1} ketma-ketligini aniqlashi va chiqarishi kerak. Har doim aynan ikkita to'g'ri javob bo'ladi (biri ikkinchisining teskarisi), va siz ulardan xohlaganini chiqarishingiz mumkin. Sizning balingiz yechimingiz ilovaga yuboradigan so'rovlar soniga bog'liq bo'ladi.

Amalga oshirish

⇒ Bu interaktiv masala. Sizning dasturingiz quyida tavsiflangan formatda tekshiruvchi dastur (grader) bilan muloqot qilish uchun standart kiritish va chiqarishdan foydalanadi.

Dasturingiz dastlab bitta musbat butun T sonini, ya'ni keyingi testlar sonini o'qishdan boshlashi kerak.

Har bir test uchun dasturingiz bitta musbat butun N sonini o'qishi kerak. Bu o'rindiqlar soni va mehmonlar sonini anglatadi.

So'rov yuborish uchun dasturingiz “? $I\ J\ K$ ” ko'rinishidagi qatorni chiqarishi kerak, bu yerda $0 \leq I, J, K \leq N - 1$ — uchta **turli xil** sonlar.

So'rov yuborgandan so'ng, dasturingiz so'rovingizga javob sifatida bitta musbat butun son joylashgan qatorni o'qishi kerak.

To'g'ri o'tirish tartibi bilan javob berish uchun dasturingiz “! $g_0 \dots g_{N-1}$ ” ko'rinishidagi qatorni chiqarishi kerak.

Barcha T ta testni yechgandan keyin dasturingiz odatdagidek (xatosiz) yakunlanishi kerak.

Shuni e'tiborga olingki, CMS tizimida yechimingizni tekshirish uchun ishlatiladigan rasmiy grader **adaptiv (moslashuvchan)** bo'lishi mumkin. Ya'ni, ba'zi testlar uchun mehmonlarning o'zni oldindan belgilanmagan bo'lishi mumkin. Buning o'rniga, grader dasturingiz tomonidan berilgan so'rovlarga asoslangan holda mumkin permutatsiyalardan qaysidir birini ishlatishni o'zi hal qilishi mumkin.

Buferni tozalash (Flushing). Agar siz taqdim etilgan shablonlardan foydalanmayotgan bo'lsangiz, har bir qatorni chiqarishdan so'ng standart chiqarish buferini tozalashni unutmang, aks holda dasturingiz *Not correct* (Noto'g'ri) deb baholanishi mumkin. Pythonda, agar qatorlarni o'qish uchun `input()` dan foydalansangiz, bu avtomatik tarzda sodir bo'ladi yoki tozalashga majburlash uchun `print(..., flush=True)` dan foydalanishingiz mumkin. C++ tilida `cout << endl;` yangi qatorga o'tish bilan birga buferni ham tozalaydi; agar `printf` ishlatasangiz, `fflush(stdout)` dan foydalaning.

Cheklovlar

- $1 \leq T \leq 10$.
- N qiymati 5 (faqat misol uchun), 8, 40 yoki 2000 bo'ladi.
- Har bir test uchun siz ko'pi bilan 10 000 ta so'rov yuborishingiz mumkin.

Baholash

Dasturingiz qism-masalalarga guruhlangan (subtask) bir nechta testlarda sinovdan o'tkaziladi. Biror qism-masaladan ball olishingiz uchun, sizning dasturingiz undagi barcha testlarda to'g'ri ishlashi kerak.

- **Qism-masala 0 [0 ball]:** Sample misol ($N = 5$).
- **Qism-masala 1 [9 ball]:** $N = 8$.
- **Qism-masala 2 [11 ball]:** $N = 2000$, va 0- va 1-mehmonlar yonma-yon o'tirishadi.
- **Qism-masala 3 [15 ball]:** $N = 40$.
- **Qism-masala 4 [65 ball]:** $N = 2000$.

1 va 2-qism-masalalar uchun barcha testlarni to'g'ri yechgan har qanday yechimga to'liq ball beriladi.

3 va 4-qism-masalalarda biron-bir ball olish uchun yechimingiz barcha testlarni to'g'ri yechishi shart, va sizning balingiz bitta testni yechish uchun ketgan eng ko'p so'rovlar soni — Q_s ga bog'liq bo'ladi. Aytaylik, $X_s = \max(1, Q_s/N)$. Unda 3 va 4-qism-masalalar uchun ballar quyidagicha hisoblanadi:

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

score_s qiymati har bir qism-masala uchun eng yaqin butun songacha yaxlitlanadi, va sizning umumiy balingiz shularning yig'indisiga teng bo'ladi. To'liq ball olish uchun 3-qism-masalani ko'pi bilan 55 ta, 4-qism-masalani esa ko'pi bilan 2597 ta so'rov bilan yechishingiz kerak. Q_s ning namunaviy qiymatlari va 3 hamda 4-qism-masalalar uchun ballar quyida keltirilgan.

| | | | | | | | | |
|------------------|----|----|----|----|----|-----|-----|-------|
| 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 |

Misollar

| Grader | Yechim |
|--------|-------------|
| 1 5 | |
| | ? 0 2 4 |
| 3 | |
| | ? 3 0 1 |
| 3 | |
| | ? 0 4 3 |
| 5 | |
| | ! 3 1 0 2 4 |

Izoh

Namunaviy kiritish ma'lumotlarida $N = 5$ ta mehmon qatnashgan bitta test ($T = 1$) mavjud. Ushbu testdagi yashirin mehmonlar konfiguratsiyasi Figure 1 ga mos keladi.

Namunaviy yechim tomonidan berilgan birinchi so'rov 0, 2, 4. Bu so'rovga qaytarilgan 3 javobi ushbu mehmonlar noma'lum tartibda ketma-ket joylashgan uchta o'rindiqla yonma-yon o'tirishlarini bildiradi.

Ikkinchi so'rovga qaytarilgan 3 javobi 3, 0 va 1-mehmonlar haqida ham xuddi shunday ma'lumot beradi.

Endi biz shunday xulosaga kelishimiz mumkinki, 0-mehmon o'rtada, 2 va 4-mehmonlar uning bir tomonida, 1 va 3-mehmonlar esa ikkinchi tomonida o'tirishi kerak.

Uchinchi so'rovdan so'ng, mehmonlar [3, 1, 0, 2, 4] yoki uning teskarisi [4, 2, 0, 1, 3] tartibida o'tirishiga ishonch hosil qila olamiz. Biz ikkala tartibdan xohlaganimizni chiqarishimiz mumkin.

CMS tizimidagi kod shablonlari va baholash tafsilotlari

C++ va Python uchun taqdim etilgan kod shablonlaridan foydalanishni qat'iy tavsiya qilamiz. Ular grader bilan muloqot muvaffaqiyatli bo'lganini tekshiradi va agar muvaffaqiyatsiz bo'lsa, xatosiz to'xtaydi.

Agar berilgan shablonlardan foydalanmasangiz, yechimingiz noto'g'ri bo'lgan hollarda CMS noto'g'ri verdikt ko'rsatishi mumkin. Masalan, "Output isn't correct" o'rniga "Execution killed by signal" yoki "Execution timed out (wall clock limit exceeded)" verdiktini olishingiz mumkin.

Shuningdek, yechimingizni topshirishdan oldin uni mahalliy ravishda sinab ko'rish uchun testlash vositasidan (quyiga qarang) foydalanishni tavsiya qilamiz. Testlash vositasi yechimingizning chiqishlarini tekshiradi va noto'g'ri so'rovlardan foydalanilgan hollarni xabar qiladi.

Testlash vositasi (Testing Tool)

Yechimingizni tekshirishni osonlashtirish uchun biz CMS dan yuklab olishingiz mumkin bo'lgan oddiy vositani taqdim etamiz. Ushbu vositadan foydalanish ixtiyoriy. E'tibor bering, CMS dagi rasmiy grader testlash vositasidan farq qiladi.

Vositadan foydalanish uchun sizga kiritish (input) fayli kerak bo'ladi. Taqdim etilgan `seatingplan.input0.txt` namunaviy kiritishidan foydalanishingiz yoki o'zingiznikini yaratishingiz mumkin. Kiritish fayli testlar soni T ko'rsatilgan qator bilan boshlanishi kerak, shundan so'ng u har bir test uchun ikkita qatordan iborat bo'lishi lozim: N soni yozilgan birinchi qator va g_0, g_1, \dots, g_{N-1} sonlari yozilgan ikkinchi qator.

`seatingplan.py` deb nomlangan Python dasturlari uchun (odatda `pypy3 seatingplan.py` sifatida ishga tushiriladi) testlash vositasini quyidagicha ishlating:

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

C++ dasturlari uchun avval yechimingizni kompilyatsiya qiling:

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

va shundan so'ng testlash vositasini ishga tushiring:

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