

Kāršu triks

Divi spēlētāji gatavojas demonstrēt kāršu triku ar standarta 52 kāršu komplektu. Ērtības labad katrai kārtij ir piešķirts skaitlis no 0 līdz 51.

Kārtis sākotnēji tiek novietotas uz galda vienā rindā, ar vērtībām uz augšu kaut kādā spēlētājiem iepriekš nezināmā secībā.

Pirms trika abiem spēlētājiem tiek paziņoti divi skaitļi: S un T.

Pirmais spēlētājs iet pie galda, paskatās kārtis un veic kāršu maiņu, kopā ne vairāk kā S reizes. Katra kāršu maiņa tiek veikta, izvēloties divas kārtis i un j pozīcijās (i un j var būt vienādas) un pārvietojot kārti no pozīcijas i uz pozīciju j un otrādi.

Pēc tam pirmais spēlētājs aiziet, nesazinoties ar otro spēlētāju, un visas kārtis tiek apgrieztas (to vērtības vairs nav redzamas), nemainot to kārtību. Otrais spēlētājs tiek uzaicināts pie galda un tiek lūgts uzminēt, kur atrodas kārts ar noteiktu vērtību. Viņam ir atļauts atklāt pa vienai ne vairāk kā T kārtis. Ja kāda no atklātajām kārtīm ir meklētā, tad spēlētāji uzvar. Ja kārti ar T minējumiem neizdodas atrast, viņi zaudē.

Jūsu mērķis ir uzrakstīt divas programmas, kas imitēs spēlētāju darbības un uzvarēs spēlē.

Risināšanas detaļas

Jums tiks dotas divas programmas - *FirstPlayer* un *SecondPlayer* kopā ar paraugvērtētāju.

Programmā *FirstPlayer* jums ir jāizveido šāda funkcija:

```
void swapCards(int cards[], int S, int T)
```

- Šo funkciju vērtētājs izsauc tieši vienu reizi
- cards: masīvs, kas satur sākotnējās kāršu vērtības no kreisās uz labo pusi, ar precīzi 52 elementiem, kas indeksēti no 0 līdz 51
- S: kāršu maiņu skaits
- T: atļauto minējumu skaits

swapCards var izsaukt sekojošu funkciju:

```
void doSwap(int i, int j)
```

- i: pirmās apmaināmās kārts indekss, $0 \leq i < 52$
- j: otrās apmaināmās kārts indekss, $0 \leq j < 52$
- **doSwap** var izsaukt ne vairāk kā S reizes

Programmā *SecondPlayer* jums ir jāizveido šāda funkcija:

```
void guessCard(int S, int T, int target)
```

- S: kāršu maiņu skaits
- T: atļauto minējumu skaits
- target: meklējamās kārts vērtība

guessCard var izsaukt sekojošu funkciju:

```
int guess(int idx)
```

- idx: pārbaudāmās kārts indekss, $0 \leq idx < 52$
- šī funkcija atgriež idx-tās kārts vērtību
- guess var izsaukt ne vairāk kā T reizes
- pareiza minējuma gadījumā meklēšana tiek veiksmīgi pabeigta

Piemērs

Tabulā ir redzams ievades datu paraugs vērtētājam un funkciju izsaukumu apraksts šim piemēram.

Pirmajā ievades datu rindā jābūt diviem skaitļiem: S un T.

Otrajā ievades datu rindā jābūt 52 veseliem skaitļiem, kur i-tais skaitlis apzīmē i-tās kārts vērtību.

Trešajā ievades datu rindā ir meklējamās kārts vērtība (**target**).

| Ievades datu paraugs vērtētājam | Funkciju izsaukumu piemēri | | |
|---------------------------------|-----------------------------|-----------------|------------------------------------|
| | Izsaukumi | Apakš-izsaukumi | Atgriešana |
| 1 51 | swapCards([0,1,...], 1, 51) | | |
| 0 1 2 3 4 5 6 7 8 | | doSwap(0, 1) | |
| 9 10 11 12 13 14 | | | Apmaina kārtis ar indeksiem 0 un 1 |
| 15 16 17 18 19 | | | |
| 20 21 22 23 24 | swapCards ir pabeigta | | . |
| 25 26 27 28 29 | guessCard(1, 51, 1) | | |
| 30 31 32 33 34 | | guess(5) | |
| 35 36 37 38 39 | | | guess atgriež 5 |
| 40 41 42 43 44 | | guess(1) | |
| 45 46 47 48 49 | | | guess atgriež 0 |
| 50 51 | | guess(0) | |
| 1 | | | Pareizi! |

Ierobežojumi

- $1 \leq S \leq 52$
- $1 \leq T \leq 51$
- $0 \leq target < 52$

Apakšuzdevumi

1. (16 punkti): $S = 52, T = 1$
2. (20 punkti): $S + T = 52$
3. (22 punkti): $S = 13, T = 27$
4. (18 punkti): $S = 1, T = 26$
5. (24 punkti): Dotajām S un T eksistē uzvaroša stratēģija