

(PL) Sztuczka karciana

EJOI 2020, dzień 2
5.09.2020

Kod zadania: **cards**
Limit czasu: **0.5 s**
Limit pamięci: **512 MB**



Dwóch graczy chce zaprezentować sztuczkę karcianą używając standardowej talii 52 kart. Dla ułatwienia, kolejne karty będziemy oznaczali kolejnymi liczbami naturalnymi od 0 do 51 włącznie. Karty są początkowo ułożone na stole w jednym rzędzie z widocznymi oznaczeniami w jakiejś kolejności, która jest nieznana przez graczy.

Pierwszy gracz siada do stołu, patrzy na karty i wykonuje co najwyżej S zamian. Każda zamiana polega na wybraniu dwóch pozycji i oraz j (i oraz j mogą być równe) i zamienieniu kolejnością kart na tych pozycjach: karta z pozycji i łąduje na pozycji j i na odwrót.

Następnie pierwszy gracz odchodzi od stołu bez komunikacji z drugim graczem i karty zostają odwrócone bez zmieniania ich kolejności. Drugi gracz siada do stołu i ma odgadnąć, gdzie znajduje się karta z oznaczeniem *target*. W tym celu wolno mu odwrócić co najwyżej T kart jedna za drugą. Jeśli którakolwiek z tych kart to *target*, wtedy gracze wygrywają i sztuczka się udała. Jeśli jednak drugi gracz wykorzysta wszystkie odwrócenia i nie znajdzie szukanej karty wtedy gracze przegrywają.

Twoim celem jest napisać dwa programy, które symulują ruchy dwóch graczy, żeby wygrać grę.

Interakcja

Masz do dyspozycji dwa programy `FirstPlayer` oraz `SecondPlayer` wraz z przykładową sprawdzaczką.

W programie `FirstPlayer` masz zaimplementować następującą funkcję:

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

Funkcja ta będzie uruchomiona dokładnie raz przez sprawdzaczkę. Parametr `cards` jest tablicą długości 52 zawierającą początkowe, parami różne wartości kart od 0 do 51, parametr S jest liczbą dopuszczalnych zamian, zaś parametr T jest liczbą kart, które będzie mógł odwrócić drugi gracz.

Powyższa funkcja może wywoływać następującą funkcję:

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

Parametr i oznacza pozycję pierwszej karty do zamiany ($0 \leq i < 52$), a parametr j oznacza pozycję drugiej karty do zamiany ($0 \leq j < 52$). Funkcja może być wywołana co najwyżej S razy.

W programie `SecondPlayer` masz zaimplementować następującą funkcję:

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

Parametr S jest liczbą dopuszczalnych zamian, parametr T jest liczbą kart, które może odwrócić drugi gracz, a `target` jest oznaczeniem na karcie, która powinna zostać odwrócona.

Powyższa funkcja może wywoływać następującą funkcję:

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

Parametr `idx` jest pozycją karty, którą odwraca drugi gracz ($0 \leq \text{idx} < 52$). Zwraca oznaczenie karty na pozycji `idx`. Funkcja może być wywołana co najwyżej T razy. Gdy program odgadnie właściwie, uruchomienie programu zakończy się.

Przykładowa interakcja

Poniżej znajduje się przykład wejścia dla dołączonej sprawdzaczki. W pierwszym wierszu wejścia znajdują się dwie liczby naturalne S i T . W drugim wierszu znajdują się 52 liczby: i -ta liczba to oznaczenie i -tej karty. W trzecim wierszu znajduje się liczba `target`.

Przykładowe wejście sprawdzaczki	Przykładowe wywołania		
	Wywołanie	Podwywołanie	Zwrócona odpowiedź
1 51	swapCards([0, 1, ...], 1, 51)		
0 1 2 3 4 5 6 7 8		doSwap(0, 1)	
9 10 11 12 13			zamienione zostały karty na pozycjach 0 i 1
14 15 16 17 18			
19 20 21 22 23	swapCards kończy się		
24 25 26 27 28	guessCard(1, 51, 1)		
29 30 31 32 33		guess(5)	
34 35 36 37 38			guess zwraca 5
39 40 41 42 43		guess(1)	
44 45 46 47 48			guess zwraca 0
49 50 51		guess(0)	
1			Dobrze!

Ograniczenia

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

Podzadania

Zestaw testów dzieli się na podane poniżej podzadania.

Dodatkowe ograniczenia	Liczba punktów
$S = 52, T = 1$	16
$S + T = 52$	20
$S = 13, T = 27$	22
$S = 1, T = 26$	18
Strategia wygrywająca dla graczy istnieje dla podanych S i T	24