

Zadanie: PKN

Papier, kamień, nożyce [A]



POTYCZKI ALGORYTMICZNE

Potyczki Algoritmiczne 2026, runda druga. Limity: 1024 MB, 5 s.

2026-03-24

Algosi i Bajtek znaleźli się w bardzo nietypowej sytuacji. Każde z nich posiada swój własny ciąg binarny długości n . Chcieliby się tymi ciągami jak najszybciej wymienić.

Sprawę utrudnia fakt, że aktualnie przebywają oni na mistrzostwach gry w papier, kamień, nożyce. Konkurs ten ostatnio przeszedł technologiczną rewolucję. Aby uniknąć jakichkolwiek prób komunikacji między uczestnikami i w pełni skupić się na losowości strategii zawodników, organizatorzy postanowili całkowicie odizolować od siebie uczestników i postawić między nimi system komputerowy. Każdy z uczestników deklaruje swój ruch i dopiero po tym odsłaniane są wyniki rundy.

Zasady meczu w papier, kamień, nożyce są następujące:

- Mecz składa się z rund.
- Podczas pojedynczej rundy każdy z graczy wybiera jeden z trzech symboli: papier, kamień lub nożyce.
- Następnie symbole te są porównywane. Jeśli gracze wybrali ten sam symbol, to runda kończy się remisem i nikt nie zdobywa punktu. W przeciwnym przypadku osoba z silniejszym symbolem zdobywa 1 punkt (papier wygrywa z kamieniem, kamień wygrywa z nożycami, a nożyce wygrywają z papierem).
- Gracz, który jako pierwszy będzie miał o 2 punkty więcej niż przeciwnik, wygrywa mecz.
- Rundy trwają, dopóki któryś z graczy nie wygra meczu.

Algosi i Bajtkowi zależy na tym, aby nawzajem poznać swoje słowa jeszcze przed końcem meczu. Chcieliby też grać jak najmniej rund, żeby nie nadwyreżać cierpliwości obserwatorów i kibiców. Napisz program, który im w tym pomoże. Musisz rozwiązać ten problem dla t niezależnych przypadków testowych.

Interakcja

To jest zadanie interaktywne. Twój program zostanie uruchomiony w dwóch kopiach – jednej dla Algosi, a drugiej dla Bajtka. Każde z uruchomień w pierwszym wierszu wejścia otrzyma słowo **Algosi** lub **Bajtek**, które określa za działanie której z osób odpowiada ta kopia programu.

Format wejścia w obu przypadkach wygląda identycznie. Pierwszy wiersz wejścia zawiera słowo **Algosi** lub **Bajtek**. Drugi wiersz zawiera dwie liczby n i t ($1 \leq n \leq 5000, 1 \leq t \leq 5$), określające kolejno długość ciągów binarnych, jakie chcą między sobą przesłać Algosi i Bajtek oraz liczbę przypadków testowych. Następnie t razy odbywa się interakcja między programami.

W pierwszym wierszu każdego przypadku testowego obydwa programy otrzymają napis długości n , składający się tylko ze znaków 0 i 1. Po wczytaniu swoich napisów Algosi i Bajtek zaczynają rozgrywkę. Podczas pojedynczej rundy gracz najpierw wypisuje wiersz zawierający jego ruch, określany jednym znakiem $c \in \{P, K, N\}$, a następnie wczytuje z wejścia wiersz zawierający znak określający ruch drugiego gracza w tym samym formacie. Maksymalna liczba odbytych rund w pojedynczym przypadku testowym wynosi 20000. Przekroczenie tego limitu będzie skutkowało werdyktem „błędna odpowiedź”.

Aby zakończyć przypadek testowy, należy wypisać wiersz zawierający znak ! (wykrzyknik), odstęp oraz następujący po nim napis długości n składający się tylko ze znaków 0 i 1. Dla Algosi powinien być to napis Bajtka, zaś dla Bajtka napis Algosi. Po wypisaniu wyniku program powinien od razu przejść do kolejnego przypadku testowego (czyli wczytać wiersz zawierający napis, który należy przekazać) lub zakończyć działanie, jeśli był to ostatni przypadek testowy.

Po wypisaniu odpowiedzi należy opróżnić bufor wyjścia, na przykład poprzez wywołanie polecenia `cout.flush()` lub `fflush(stdout)`.

Przykładowa interakcja

Interaktor do Algosi	Algosi do interaktora	Interaktor do Bajtka	Bajtek do interaktora	Wyjaśnienie
Algosi		Bajtek		Interaktor informuje kopie programów Algosi i Bajtka o ich tożsamości.
5 2		5 2		Oboje się dowiadują, że będą dwa przypadki testowe i w każdym z nich napisy do przekazania będą miały 5 znaków.
10010		00001		Algosi i Bajtek poznają swoje napisy. Przykładowo Bajtek ma przekazać Algosi cztery zera i jedynekę.
	P		K	Zaczynają grać. W pierwszej rundzie Algosi gra papier, a Bajtek – kamień. Algosi zdobywa jeden punkt!
K		P		Oboje dowiadują się, co zagrał drugi gracz.
	P		P	Algosi wie, że nie może teraz wygrać, bo mecz się skończy. Żeby ułatwić Bajtkowi zadanie, gra ponownie papier. Bajtek faktycznie domyślił się, co się dzieje, też zagrał papier. Algosi wciąż ma o jeden punkt więcej.
P		P		Oboje poznają zagranie drugiego gracza, i każde z nich decyduje, że już pora zgadywać.
	! 00001		! 10010	To faktycznie poprawne odpowiedzi w pierwszym przypadku testowym.
00010		10001		Interaktor podaje każdemu z graczy napis do przekazania.
	P		K	Algosi zdobywa punkt! Nie kończy to jeszcze gry, bo punkty w każdym przypadku testowym są naliczane od nowa.
K		P		Oboje dowiadują się, co zagrał drugi gracz.
	! 10001		! 00010	Aby nieco skrócić ten opis (oraz nie ryzykować wygranej Algosi) decydują się zgadywać. Uczestnikom Potyczek jury poleca przed zgadywaniem przekazać nieco więcej informacji.

Uwagi

- Opisana powyżej interakcja odpowiada pierwszemu testowi przykładowemu (o nazwie 0a).
- We wszystkich testach pozostałych testach (również w drugim teście przykładowym o nazwie 0b) zachodzi $t = 5$, $n = 5000$.
- Jeśli po rundzie któryś z graczy ma o 2 punkty więcej niż przeciwnik, program od razu kończy się z werdyktem „błędna odpowiedź”.
- Obydwa programy rozpoczynają się w tym samym czasie. Czas działania programu mierzony jest jako czas rzeczywisty od rozpoczęcia do zakończenia interaktora.
- Przewagi nie przenoszą się między przypadkami testowymi. Na początku każdego przypadku obaj gracze zaczynają z 0 punktami (niezależnie od tego, z jakim wynikiem skończył się poprzedni przypadek).

Ocenianie

Zestaw testów składa się z jednej grupy, za którą można dostać maksymalnie 10 punktów. Niech m oznacza maksymalną liczbę odbytych rund po wszystkich przypadkach testowych pojedynczego testu. Wynik za dany test określany jest według następujących zasad:

$m \leq$	Liczba punktów
20000	1
16250	2
10000	3
8750	4
6250	5
5500	6
5250	7
5125	8
5050	9
5000	10

Końcowy wynik zgłoszenia to minimalny wynik po wszystkich testach.

Testowanie lokalne

W sekcji *Pliki* dostępny jest interaktor `pksoc.cpp`. Jest on identyczny jak interaktor używany do sprawdzania zgłoszeń na SIO2. Należy go uruchamiać komendą:

```
python3 pknrun.py [rozwiązanie] < [test] > [wyjście]
```

przy czym pliki `pksoc.cpp` oraz `oi.h` muszą znajdować się w tym samym folderze. Format, w którym interaktor przyjmuje wejście, jest opisany w komentarzu w pliku `pksoc.cpp`.

Zasady fair play

Komunikowanie się pomiędzy programami w inny sposób niż za pomocą przebiegu rozgrywki, np. poprzez wysłanie ruchu przez jeden program z opóźnieniem i odczytanie zegara przez drugi jest zabronione, w przypadku stwierdzenia przez Jury próby komunikowania się między programami w niedozwolony sposób zgłoszenie może zostać usunięte, a w rażących przypadkach może zostać podjęta decyzja o dyskwalifikacji z całego konkursu.