

(PL) Triangulacja

EJOI 2020, dzień 1
4.09.2020

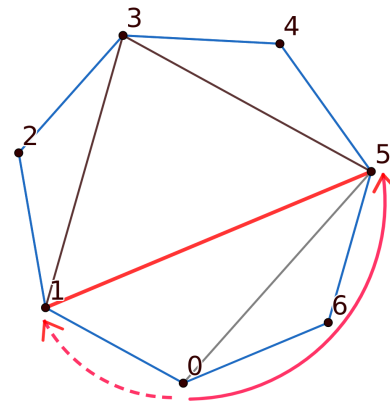
Kod zadania: **triangulation**
Limit czasu: **1 s**
Limit pamięci: **512 MB**



Ania narysowała n -kąt foremny, którego wierzchołki ponumerowała od 0 do $n-1$ zgodnie z kierunkiem wskazówek zegara. Następnie narysowała triangulację tego wielokąta czyli $n-3$ nieprzecinających się przekątnych. Przekątna to oczywiście odcinek między dwoma wierzchołkami, które nie są połączone bokiem wielokąta.

Definiujemy teraz *odległość* od wierzchołka A do przekątnej D . Załóżmy, że rozpoczynamy spacer od wierzchołka A i poruszamy się do kolejnych wierzchołków zgodnie z ruchem wskazówek zegara aż napotkamy jeden z końców przekątnej D . Liczba przebytych boków to *odległość w lewo*. Analogicznie, *odległość w prawo* to liczba boków, które przebywa się spacerując od A do jednego z końców przekątnej D spacerując przeciwnie do ruchu wskazówek zegara. Przyjmujemy, że zawsze należy przebyć co najmniej jeden bok (odległość w lewo i odległość w prawo nigdy nie jest 0). Odległość od A do D to **maksimum** z odległości w lewo oraz odległości w prawo.

W przypadku obok odległość między wierzchołkiem 0 a przekątną (1, 5) jest równa 2, ponieważ lewa odległość jest równa 1, a prawa odległość 2. Odległość między wierzchołkiem 0 a przekątną (0, 5) jest natomiast równa 5, ponieważ lewa odległość jest równa 5, a prawa odległość 2.



Ania rzuciła wyzwanie Jakubowi. Jakub nie wie, które przekątne zostały narysowane w triangulacji Ani. Zna jedynie n i może zadawać Ani pytania o pewne pary wierzchołków i ona odpowie czy istnieje przekątna pomiędzy tymi wierzchołkami. Celem Jakuba jest znaleźć przekątną, której odległość (zdefiniowana powyżej) od wierzchołka 0 jest najmniejsza. Twoim celem jest pomóc Jakubowi osiągnąć cel zadając Ani możliwie mało pytań.

Ograniczenia

$$5 \leq n \leq 100$$

Implementacja

Należy napisać poniższą funkcję:

```
int solve(int n)
```

Funkcja będzie uruchomiona przez sprawdzaczkę dokładnie raz. Parametr n oznacza liczbę wierzchołków wielokąta. Funkcja powinna zwrócić najbliższą przekątną (a, b) zakodowaną jako pojedynczą liczbę $a \cdot n + b$. Jeżeli istnieje wiele poprawnych odpowiedzi, program może zwrócić dowolną z nich.

Powyższa funkcja może wywoływać funkcję:

```
int query(int x, int y)
```

Parametry x i y oznaczają numery wierzchołków ($0 \leq x, y \leq n-1$). Funkcja zwraca 1, jeżeli istnieje przekątna między wierzchołkami x oraz y oraz 0 w przeciwnym przypadku.

Przykładowa interakcja

Poniżej znajduje się przykładowe wejście dla sprawdzaczki i przykładowe wywołania zgodne z rysunkiem powyżej. W pierwszym (i jedynym) wierszu wejścia dla sprawdzaczki znajduje się jedna liczba naturalna n .

Przykładowa sprawdzaczka będzie wypisywać każde zapytanie query na wyjście i oczekiwać od Ciebie żeby na to zapytanie odpowiedzieć 1 lub 0.

Przykładowe wejście sprawdzaczki	Przykładowe wywołania			
	Wywołanie	Zwrócona odpowiedź	Wywołanie	Zwrócona odpowiedź
7	solve(7)			
			query(0, 3)	
				query zwróciło 0
			query(0, 5)	
				query zwróciło 1
			query(1, 5)	
				query zwróciło 1
		solve zwraca $1 \cdot 7 + 5 = 12$		
	Dobrze!			

Ocenianie

Niech q oznacza liczbę zapytań, które zadał Twój program na pojedynczym teście. Ponadto, niech $w = \frac{n \cdot (n-3)}{2}$.

- Jeśli program zada niewłaściwe pytanie lub odgadnie niewłaściwie, otrzymasz 0% punktów za test.
- Jeśli $w < q$, otrzymasz 0% punktów za test.
- Jeśli $n < q \leq w$, otrzymasz $(10 + 60 \cdot \frac{w-q}{w-n})\%$ punktów za test.
- Jeśli $q \leq n$, otrzymasz 100% punktów za test.

Podzadania

Jest tylko jedno podzadanie. Twój wynik jest sumą za poszczególne testy. Podczas zawodów zobaczysz wyniki tylko dla połowy testów wartej 50 punktów. Druga połowa wyników testów zostanie ujawniona dopiero po konkursie. Twój ostateczny wynik zadania jest najwyższym łącznym wynikiem pojedynczego zgłoszenia.