

Triangulacija

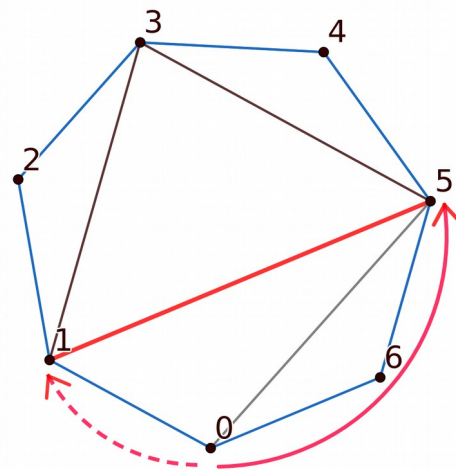
Opis problema

Anna je nacrtala pravilni poligon sa n vrhova numerisanih od 0 do $n-1$ u smjeru kazaljke na satu. Kasnije ga je triangulirala dodajući nekih $n-3$ dijagonala koje se međusobno ne sijeku, to jest nemaju zajedničkih tačaka osim onih krajnjih tačaka. Jedna dijagonala se definiše kao prava linija između dva različita vrha koja nisu na istoj stranici poligona.

Prvo, definišimo udaljenost od vrha A do dijagonale D . Pretpostavimo da započnemo od vrha A i nastavimo se kretati prema sljedećem vrhu u smjeru kazaljke na satu dok ne dođemo do jedne od krajnjih točaka dijagonale D . Broj pređenih stranica nazvat ćemo **left_distance**. Na sličan način, **right_distance** je broj pređenih stranica ako započnemo od A i krećemo se u smjeru suprotnom od smje $right_distance$ ra kazaljke na satu dok ne dođemo do D . Udaljenost od A do D je maksimum od **left_distance** i **right_distance**.

U primjeru na datoj slici udaljenost od vrha 0 do dijagonale (1,5) je 2, pri čemu je **left_distance** 1, a **right_distance** 2. Na sličan način, udaljenost između dijagonale (0,5) i vrha 0 je 5 jer je **left_distance** = 5 i **right_distance** = 2 pa je maksimum 5.

Anna sada želi postaviti jedan izazovan zadatak za Jacoba. Jakob uopšte ne zna koje dijagonale su nacrtane na poligonu. Zna samo vrijednost broja n , ali može Annu više puta pitati o nekim parovima vrhova i ona će mu reći postoji li ili ne dijagonala između tih vrhova. Jacobov cilj je pronaći najbližu nacrtanu dijagonalu od vrha 0 (s udaljenošću definiranom kao gore). Vaš zadatak je da pomognete Jacobu da postigne svoj cilj postavljajući Ani ograničen broj pitanja.



Ograničenja

- $5 \leq n \leq 100$

Detalji implementacije

Vi treba da implementirate sljedeću funkciju kao dio vašeg rješenja:

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

- Ova funkcija se poziva tačno jednom od strane gradera
- n : broj vrhova poligona
- Ova funkcija treba da vrati dijagonalu između nekog para vrhova a i b u obliku cijelog broja vrijednosti $a \cdot n + b$
- Ukoliko ima više dijagonala na minimalnoj udaljenosti možete vratiti bilo koju od njih

Gore definisana funkcija može pozivati sljedeću funkciju:

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

- x : broj na prvom vrhu
- y : broj na drugom vrhu
- $0 \leq x, y \leq n - 1$
- vraća 1 ako postoji dijagonala između vrhova x i y a 0 inače

Primjer izvršenja

Ovdje dolje je opisan primjer ulaza u program za ocjenjivanje (grader) i poziva napravljenih tokom izvršenja funkcije. Ovi pozivi odgovaraju i opisani su na datoj slici gore u tekstu.

Jedina linija inputa ima jedan cijeli broj: n

Primjer program za ocjenjivanje će isprintati svaki poziv na standardni output stdout i vi morate ručno odgovoriti na pitanje sa 1 ili 0 (da ili ne).

Primjer ulaza u program za ocjenjivanje	Primjer poziva funkcija			
	pozivi	vraća	pozivi	vraća
7	solve(7)			
			query(0, 3)	
				query vraća 0
			query(0, 5)	
				query vraća 1
			query(1, 5)	
				query vraća 1
		solve vraća $1 \cdot 7 + 5 = 12$		
		TAČNO!		

Bodovanje

Označimo sa q broj pitanja koja ste koristili tokom jednog testa. Pored vrijedi da je $w = \frac{n \cdot (n - 3)}{2}$.

- Ako pogrešno postavite pitanje ili je vaš odgovor pogrešan dobićete 0% bodova za test
- Ako je $w < q$ dobićete 0% bodova za test
- Ako je $n < q \leq w$ dobićete $10 + 60 \cdot \frac{w - q}{w - n} \%$ bodova za test
- Ako je $q \leq n$ dobićete 100% bodova za test

Podzadaci

Ima samo jedan podzadatak i ukupan broj bodova je zbir dobivenih bodova po svim testovima. S druge strane, tokom takmičenja moći ćete da vidite samo rezultate, odnosno broj bodova za polovinu testova (koji zajedno vrijede 50 bodova). Druga polovina bodova koju imate na zadatku ćete znati tek nakon takmičenja. Vaš krajnji ukupan broj bodova će biti **najbolji postignuti rezultat od svih rješenja koja ste predali**.