

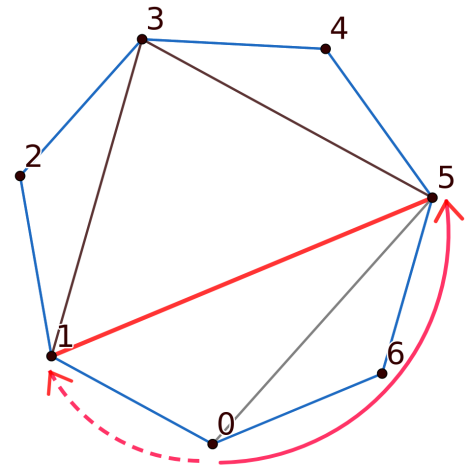
Üçgenlere ayırma

Problem tanımı

Anna n köşeden oluşan ve köşelerin saat yönünde 0'dan $n - 1$ 'e numaralandırıldığı düzgün bir çokgen çizer. Daha sonra birbirleri ile kesilmeyen $n - 3$ tane köşegen çizerek bu çokgeni üçgenlere ayırır. Köşegenler uç noktalarında birbirlerine değebilirler. Köşegen, ortak bir kenara değmeyen farklı iki köşe arasında çizilen doğru parçasıdır.

Bir A köşesinden D köşegenine olan uzaklığı şu şekilde tanımlayalım. A köşesinden başlayıp saat yönünde köşeleri sırayla gezerek D köşegeninin bir uç noktasına varana kadar ilerlediğimizi düşünelim. Bu ilerleme sırasında gezilen kenar sayısına **sol_uzaklık** diyelim. Benzer şekilde, **sağ_uzaklık** da A'dan başlayıp D'ye ulaşana kadar saat yönünün tersinde gittiğimiz zaman gezilen kenar sayısı olsun. A'dan D'ye olan **uzaklık**, **sol_uzaklık** ve **sağ_uzaklık**'ın maksimumudur.

Örnek resimde 0 köşesinden (1,5) köşegenine olan uzaklık 2'dir (sol_uzaklık 1 ve sağ_uzaklık 2 olduğu için). 0 köşesinin (0,5) köşegenine olan uzaklığı ise 5'tir, sol_uzaklık=5 ve sağ_uzaklık=2.



Anna, Jacob'a bir challenge vermek ister. Jacob hangi köşegenlerin çizildiğini hiç bilmemektedir. Sadece çokgenin kaç köşesi olduğunu, n 'yi, bilir ama Anna'ya istediği kadar köşe ikilisi için o köşeler arasında köşegen olup olmadığını sorabilir ve Anna da ona o çift arasında köşegen olup olmadığını söyler. Jacob'ın hedefi, yukarıda tanımlandığı şekilde, 0 (sıfır) köşesine en yakın olarak çizilen köşegeni bulmaktır. Onun bu hedefine ulaşması için Anna'ya sınırlı sayıda sorular sorarak yardım edeceksiniz.

Kısıtlar

- $5 \leq n \leq 100$

Implementasyon detayları

Gönderdiğiniz submission'da aşağıdaki fonksiyonu implement etmelisiniz:

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

- Bu fonksiyon değerlendirici tarafından tam olarak bir kez çağrılır
- n : çokgenin köşe sayısı
- Bu fonksiyon, çözüm olan köşegeni, a ve b köşesi arasında ise, tek bir tamsayı olarak, $a \cdot n + b$ şeklinde hesaplayarak, dönmelidir.
- Eğer en küçük uzaklığa sahip birden fazla köşegen varsa bunlardan herhangi birini dönebilirsiniz.

Yukarıdaki fonksiyon aşağıdaki fonksiyonu birden fazla çağırabilir:

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

- x : ilk köşenin numarası
- y : ikinci köşenin numarası
- $0 \leq x, y \leq n - 1$
- Eğer x ve y arasında bir köşegen varsa 1 döner, yoksa 0 döner

Örnek etkileşim

Aşağıda grader'a verilen örnek bir girdi ve ilgili fonksiyon çağrıları verilmektedir. Bu girdi yukarıda resimde gösterilen örneğe karşılık gelmektedir.

Girdideki tek satır tek bir tamsayı içerir: n

Örnek grader her bir sorguyu stdout'a yazar ve 1 ya da 0 olarak elle kendiniz cevap vermeniz gerekir.

Grader'a örnek girdi	Örnek çağrılar			
	Çağrı	Dönen değer	Çağrı	Dönen değer
7	solve(7)			
			query(0, 3)	
				query 0 döner
			query(0, 5)	
				query 1 döner
			query(1, 5)	
				query 1 döner
		solve $1 \cdot 7 + 5 = 12$ döner		
		Doğru!		

Puanlama

q , bir test girdisi için kullandığınız sorgu sayısı olsun. Ek olarak, $w = \frac{n \cdot (n-3)}{2}$, olarak bir w tanımlayalım.

- Eğer yanlış bir sorgu yaparsanız ya da cevabınız yanlış olursa o test puanının %0'ını alırsınız
- Eğer $w < q$ ise o test puanının %0'ını alırsınız
- Eğer $n < q \leq w$ ise o test puanının $\%10 + 60 \cdot \frac{w-q}{w-n}$ ini alırsınız
- Eğer $q \leq n$ ise o test puanının %100'ünü alırsınız

Altgörevler

Tek bir altgörev vardır ve puanınız her bir test puanının toplamı olarak hesaplanır. Fakat yarışma sırasında testlerin yarısından aldığınız puanlar size gösterilecektir (50 puan değerinde). Puanların diğer yarısı yarışma bittikten sonra açıklanacaktır. Sorudan alacağınız puan **bütün submissionlarınız içinde en yüksek puanı alan submission'ın puanı olacaktır**.