



Görev Hapis (Task Prison)

⌚ 2 saniye 💾 1024 MB

Alice ve Bob haksız yere maksimum güvenli bir hapisaneye mahkûm edilmişlerdir. Şimdi kaçışlarını planlamalıdır. Bunu yapmak için mümkün olduğunca etkili bir şekilde iletişim kurabilmeleri gerekir (özellikle Alice'in Bob'a günlük bilgi göndermesi gerekmektedir). Ancak buluşamazlar ve yalnızca peçetelere yazılmış notlar aracılığıyla bilgi alışverişinde bulunabilirler. Alice her gün Bob'a yeni bir bilgi göndermek ister - 0 ile $N - 1$ arasında bir sayı. Her öğle yemeğinde Alice üç peçete alır ve her peçeteye 0 ile $M - 1$ arasında bir sayı yazar (tekrarlar olabilir) ve peçeteleri koltuğuna bırakır. Sonra, düşmanları Charly peçetelerden birini yok eder ve diğer ikisini karıştırır. Son olarak, Bob kalan iki peçeteyi bulur ve üzerlerindeki sayıları okur. Alice'in ona göndermek istediği orijinal sayıyı doğru bir şekilde çözmesi gerekir. Peçetelerde sınırlı alan vardır, bu nedenle M sabittir. Ancak, Alice ve Bob'un amacı bilgi akışını en üst düzeye çıkarmaktır, bu yüzden N 'yi olabildiğince büyük seçmekte özgürdürler. Alice ve Bob'a her biri için bir strateji uygulayarak, N değerini maksimize etmeye çalışarak yardım edin.

Implementasyon Detayları

Bu bir iletişim sorunu olduğundan, programınız burada açıklananın dışında veri paylaşmayan veya iletişim kuramayan iki ayrı yerde (biri Alice, diğeri Bob için) çalıştırılacaktır. Üç fonksiyonu uygulamanız gerekir:

```
int setup(int M);
```

Bu fonksiyon, Alice'in programınızı çalıştırmaya başlamasıyla birlikte ve Bob'un programınızı çalıştırmaya başlamasıyla birlikte birer kez çağrılacaktır. Fonksiyona M değeri verilir ve fonksiyon istenen N değerini dönmelidir. `setup`'a yapılan her iki çağrı da aynı N değerini dönmelidir.

```
std::vector<int> encode(int A);
```

Bu fonksiyon Alice'in stratejisini uygular. A 'yı kodlayan sayıyla ($0 \leq A < N$) çağrılacak ve A 'yı kodlayan üç sayı W_1, W_2, W_3 ($0 \leq W_i < M$) dönmelidir. Bu fonksiyon toplam T kez çağrılacaktır; günde bir kez (A değerleri günler arasında tekrarlanabilir).

```
int decode(int X, int Y);
```

Bu fonksiyon Bob'un stratejisini uygular. `encode` tarafından dönülen üç sayıdan ikisi belirli bir sırayla çağrılacaktır. `encode`'a girdi olarak verilen A değerini dönmelidir. Bu fonksiyon T kez çağrılacaktır (bu sayı, `encode` için yapılan T çağrıya karşılık gelir); aynı sırada olacaktırlar. `encode` için yapılan tüm çağrılar, tüm `decode` çağrılarından önce gerçekleşecektir.



Kısıtlar

- $M \leq 4300$
- $T = 5000$



Puanlama

Belirli bir alt görev için, aldığınız puanların tam puana oranına S diyelim. S , o alt görevdeki herhangi bir testte `setup` tarafından döndürülen en küçük N değerine bağlıdır. Ayrıca, S alt görev için tam puan almanız gereken N 'nin hedef değeri olan N^* değerine de bağlıdır:

- Çözümünüz herhangi bir testte başarısız olursa o zaman $S = 0$ olur.
- Eğer $N \geq N^*$ ise, $S = 1.0$ 'dir.
- Eğer $N < N^*$ ise, $S = \max \left(0.35 \max \left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0 \right)^{0.3} + 0.65 \left(\frac{N}{N^*} \right)^{2.4}, 0.01 \right)$ 'dir.



Altgörevler

Altgörev	Puan	M	N^*
1	10	700	82017
2	10	1100	202217
3	10	1500	375751
4	10	1900	602617
5	10	2300	882817
6	10	2700	1216351
7	10	3100	1603217
8	10	3500	2043417
9	10	3900	2536951
10	10	4300	3083817



Örnek

Aşağıdaki $T = 5$ örneğini ele alalım. Alice'in 0'ı kodlamak için üç eşit sayı veya 1'i kodlamak için üç farklı sayı gönderdiği bir kodlama şemasını ele alalım. Bob'un, Alice'in gönderdiği üç sayıdan herhangi ikisinden orijinal sayıyı çözebildiğini unutmayın.



Çalıştırma	Fonksiyon çağrısı	Dönen değer
Alice	setup(10)	2
Bob	setup(10)	2
Alice	encode(0)	{5, 5, 5}
Alice	encode(1)	{8, 3, 7}
Alice	encode(1)	{0, 3, 1}
Alice	encode(0)	{7, 7, 7}
Alice	encode(1)	{6, 2, 0}
Bob	decode(5, 5)	0
Bob	decode(8, 7)	1
Bob	decode(3, 0)	1
Bob	decode(7, 7)	0
Bob	decode(2, 0)	1

Örnek grader

Örnek grader için, encode ve decode çağrılarının tümü programınızın aynı çalıştırmasında olacaktır. Ek olarak, setup yalnızca bir kez çağrılacaktır (grader sisteminde olduğu gibi, her çalıştırmada bir kez olmak üzere iki kez değil).

Girdi sadece tek bir tamsayıdır - M . Ardından, setup işlevinin döndürdüğü N değerini yazdırır. Ardından, 0 ile $N-1$ arasında rastgele üretilen sayılarla ve encode 'den decode 'ye verilecek üç sayıdan rastgele seçilen ikisiyle (ve hangi sırayla) T kez encode ve decode işlemi yapılır. Çözümünüz başarısız olursa bir hata mesajı yazdırılır.