D. Təxmin Oyunu

Məsələnin adı	Guessing Game
Zaman Limiti	4 saniyə
Yaddaş Limiti	1 GB

Köhnə Lund şəhərində üzərində 0'dan $N-1$'ə nömrələnmiş N sayda ev olan küçə var. Emilya o evlərin birində yaşayır, dostları Aydan və Babək isə onun harda yaşadığını tapmaq istəyirlər. Amma Emilya onlara evin yerini bir başa demək əvəzinə oyun oynamaq qərarına gəlir. Oyun başlamazdan əvvəl Aydan və Babək müsbət bir tam ədəd K seçib bir strategiya qururlar. Lakin burdan sonra onlara kommunikasiya qurmaq qadağandır.

Oyunun özü iki mərhələdən ibarətdir. Birinci hissədə Emilya evləri ziyarət etmək üçün bir sıra seçir, və bunu elə edir ki öz evi ən axırda gəlsin. O, daha sonra Aydanla birlikdə həmin evləri bu sırada gəzir, amma bəri başdan Aydana hansı sırada gəzəcəklərini demir. Hər dəfə evin qabağından keçdikdə, əgər həmin ev Emilyanın evi deyilsə bu zaman Aydan o evin qapısına təbaşir ilə 1 və K arasında bir tam ədəd yazır. Sonuncu evə çatdıqda isə, hansı ki Emilyanın evidir, ora Emilya özü seçdiyi 1 və K arasında bir tam ədəd yazır.

Oyunun ikinci mərhələsində Babək küçədəki evləri 0'dan $N-1$'ə gəzərək evlərin üzərinə yazılmış ədədləri oxuyur. Bundan sonra o Emilyanın evini tapmaq istəyir. Onun iki təxmin haqqı var, və əgər o təxminlərdən hansısa biri düz olsa, o və Aydan qalib gəlirlər. Əks halda Emilya qalib gəlir.

Aydan və Babək üçün elə bir strategiya qura bilərsinizmi ki, onlar qalib gəlsin? Sizin strategiyanız K'dan asılı olaraq qiymətləndiriləcək (nə qədər az, o qədər yaxşı).

İmplementasiya

Bu bir neçə dəfə işlədilməklə (run olunmaqla) həll olunan məsələdir. Kodunuz birinci dəfə işləndikdə Aydanın strategiyasını quracaq. Daha sonra Babəkin strategiyasını quracaq.

Girişin ilk sətrində iki tam ədəd P və N olacaq. P ya 1 ya da 2 olacaq (birinci və ya ikinci mərhələ), N isə evlərin sayını göstərəcək. **Nümunə giriş xaric N həmişə $100\,000$'ə bərabər olacaq**.

Növbəti giriş verilənləri mərhələdən asılı olacaq:

1'ci mərhələ

Sizin proqramınız əvvəlcə bir sətirdə $K (1 \leq K \leq 1\,000\,000)$ ədədini çıxışa verməlidir.

Daha sonra, $N - 1$ dəfə girişdən bir sətirdə i $(0 \leq i < N)$ ədədini oxumalı və çıxışa bir sətirdə A_i $(1 \leq A_i \leq K)$ ədədini verməlidir. Burada A_i i'ci evin qapısına yazılmış ədədi göstərir. Emilyanın evinin indeksi xaric hər bir i indeksi tam olaraq bir dəfə qreyderin təyin etdiyi sıra ilə girişə veriləcək.

2'ci mərhələ

Proqramınız bir sətirdə N sayda tam ədəd $A_0, A_1, \ldots, A_{N-1}$ oxumalıdır.

Daha sonra bir sətirdə təxminləri, yəni iki tam ədəd s_1 və s_2 $(0 \leq s_i < N)$ çıxışa verməlidir. s_1 və s_2 bir birinə bərabər ola bilər.

İmplementasiya qeydləri

Qeyd edək ki, sizin proqramınız 2'ci mərhələ üçün işləndikdə proqram yenidən başlayacaq. Yəni bu iki mərhələ arasında dəyişənlər vasitəsi ilə məlumat saxlamaq mümkün olmayacaq.

Çıxışa verdiyiniz hər sətirdən sonra standart çıxışı flush etməyi unutmayın, əks halda sizin proqramınız "Time Limit Exceeded" ilə nəticələnə bilər.

Python dilində `print()` avtomatik flush edir. C++ dilində, `cout << endl;` çıxışa təzə sətir verməkdən əlavə həmçinin flush edir; əgər printf istifadə etsəniz `fflush(stdout)` sətrini də yazın.

Bu məsələ üçün qreyder **adaptiv** ola bilər, yəni sizin çıxışa verdiyiniz sətirlərdən asılı olaraq hərəkətlərini dəyişə bilər ki heuristik həllərin qarşısını ala bilsin. 1'ci mərhələ üçün kodunuzu bir dəfə işlədib, onun verdiyi çıxış verilənlərinə baxıb, həmin informasiyanı istifadə edib 1'ci mərhələni yenidən işə sala bilər.

Sizin proqramınız deterministik *olmalıdır*, yəni eyni giriş verilənləri ilə iki dəfə işlədildikdə hər iki halda cavabı eyni olmalıdır. Əgər proqramınızda təsadüfilik olmağını istəyirsinizsə o zaman sabit "random seed" istifadə edin. bunun üçün `srand` (C++) və ya `random.seed`'a (Python) sabit ədəd göndərə bilərsiniz. Əgər C++11'in təsadüfi ədəd yaratma sistemindən (random number generator) istifadə edirsinizsə onu yaradarkən "seed"-i təyin edin. Nümunə olaraq C++ dilində `srand(time(NULL))` istifadə edə bilməzsiniz. Əgər qreyder proqramınızın deterministik işləmədiyini anlayarsa "Wrong Answer" nəticəsi alacaqsınız.

Əgər proqramınızın işlədiyi müddətlərin (3 dəfəyə qədər) *cəmi* zaman limitini keçərsə o zaman həlliniz "Time Limit Exceeded" ilə nəticələnəcək.

Qiymətləndirmə

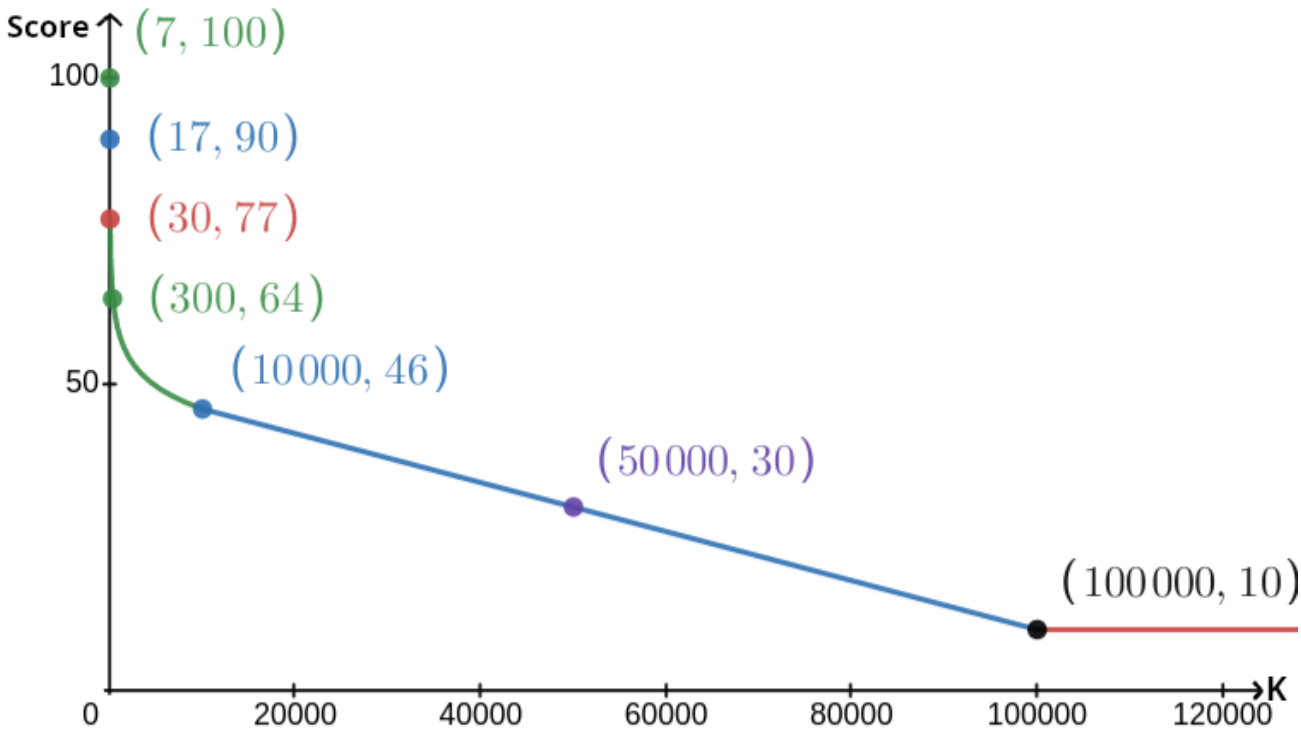
Sizin həlliniz bir neçə test üzərində yoxlanılacaq. Əgər sizin həlliniz hansısa testi keçə bilməsə (məsələn səhv cavab versə (Wrong Answer), icra xətası baş versə (Runtime Error), zaman limitini aşsa (Time Limit Exceeded), və s.), bu zaman siz 0 bal və uyğun nəticəni alacaqsınız.

Əgər sizin proqramınız *bütün* testlərdə Emilyanın evini düzgün taparsa, o zaman "Accepted" nəticəsini alacaqsınız və aldığınız bal aşağıdakı şəkildə hesablanacaq.

K_{max} sizin bütün testlərdə istifadə etdiyiniz ən böyük K olsun. Bu zaman K_{max}'dan asılı olaraq:

	Bal
$K_{max} > 99\,998$	10 bal
$10\,000 < K_{max} \leq 99\,998$	$10 + \lfloor 40(1 - K_{max}/10^5) \rfloor$ bal
$30 < K_{max} \leq 10\,000$	$46 + \lfloor 31(4 - \log_{10}(K_{max}))/(4 - \log_{10}(30)) \rfloor$ bal
$7 < K_{max} \leq 30$	$107 - K_{max}$ bal
$K_{max} \leq 7$	100 bal

Hesablama funksiyası aşağıdakı şəkildə visual olaraq göstərilib.



Hesablama zamanı nümunə hal nəzərə alınmayacaq və sizin həlliniz nümunə hal üçün işləməyə bilər.

Yoxlama aləti

Həllinizi yoxlamağı asanlaşdırmaq üçün yükləyə biləcəyiniz sadə bir alət verilir. Kattis'də məsələnin olduğu səhifənin aşağısındakı “attachments” hissəsinə baxın. Aləti dəyişə və ya istifadə etməyə də bilərsiniz. Qeyd edək ki rəsmi qreyder sizə verilən yoxlama alətindən fərqli işləyir

Nümunə istifadə ($N = 4$, $s = 2$, burada s sonuncu ziyarət olunan evin üzərinə yazılan ədəddir):

Python proqramları üçün, məsələn faylın adı `solution.py` olsun (normalda `pypy3 solution.py` sətri ilə işə salınır):

```
python3 testing_tool.py pypy3 solution.py <<<"4 2"
```

C++ proqramları üçün, əvvəlcə kompayl edin (məsələn `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) daha sonra işə salın:

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

Yoxlama aləti evləri təsadüfi sırada ziyarət edəcək. Əgər hansısa sıranı istəyirsinizsə yoxlama alətinin içində “MODIFY HERE” yazılan hissəni dəyişə bilərsiniz.

Nümunə İnteraksiya

Nümunə test qiymətləndirmə zamanı nəzərə alınmayacaq və sizin həlliniz həmin test üçün işləməyə bilər.

Fərz edək ki $N = 4$ və Emilya 1 nömrəli evdə yaşayır. Evlərin üzərində yazılan ədədləri A massivi ilə göstərək. Başda $A = [0, 0, 0, 0]$, burada 0 göstərir ki həmin evin üzərinə hələ ədəd yazılmayıb.

Sizin kodunuz birinci dəfə işlədildikdə:

$N = 4$ verilir. Sizin həlliniz $K = 3$ ilə cavab verir.

A_2 soruşulur. Sizin həlliniz 3 cavabını verir. A artıq $[0, 0, 3, 0]$ massivinə bərabərdir.

A_0 soruşulur. Sizin həlliniz 1 cavabını verir. A artıq $[1, 0, 3, 0]$ massivinə bərabərdir.

A_3 soruşulur. Sizin həlliniz 2 cavabını verir. A artıq $[1, 0, 3, 2]$ massivinə bərabərdir.

Ən sonda qreyder $A_1 = 2$ əməliyyatını yerinə yetirir və artıq A = $[1, 2, 3, 2]$ olur. Bu da birincə mərhələnin sonu olur.

Kodunuzun 2'ci mərhələsi üçün sizin həllinizə `1 2 3 2` massivi ötürülür.

Cavabınız `1 3` olur.

Təxminlərinizdən biri düz olduğu üçün (1) Aydan və Babək oyunu qazanır.

qreyderin çıxışa verdikləri	sizin çıxışa verdikləriniz
1 4	
	3
2	
	3
0	
	1
3	
	2

qreyderin çıxışa verdikləri	sizin çıxışa verdikləriniz
2 4	
1 2 3 2	
	1 3