به ازای تمام جایگشت های ۸ رقمی تعداد دور های گراف جایگشت آن را جمع می زنیم. مقدار بدست آمده چقدر است ؟

پاسخ : فرض کنید طول جایگشت n است.

بیاید به ازای هر دور ممکن بشماریم که چند بار در گراف جایگشت ها تکرار شده است. در ابتدا روی طول دور حالت بندی می کنیم. اگر دور ما به طول i باشد به $\binom{n}{i}$ طریق i راس آن را انتخاب می کنیم و با i راس به $(i-1)!$ طریق می توانیم یک دور بسازیم.

بقیه اندیس ها در جایگشت هرمقداری که باشند ما این دور به طول i را در گرافش داریم بنابر این دور ما در $(n-i)!$ جایگشت آمده است.

بنابر عبارت های بالا:

$$\binom{n}{i} \times (i-۱)! \times (n-i)! = \frac{n!}{i}$$

پس به ازای هر i باید مقدار $\frac{n!}{i}$ را جمع کنیم پس جواب به ازای $n = ۸$ برابر است با

$$8! \times \left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{8}\right) = 109,584$$

در یک ردیف n توپ داریم که رنگ هرتوپ یکی از k رنگ موجود در جهان است.
می دانیم که به ازای هر بازه متوالی از توپ ها حداقل یک رنگ وجود دارد که دقیقا یک بار در آن بازه آمده است.
به ازای $n = ۲^{۲۰۰۰۰۰} - ۱$ و $k = ۲۰۰۰۰۰$ چند دنباله از این توپ ها وجود دارد به صورتی که شرط بالا را داشته باشد ؟
باقی مانده تعداد حالات ممکن را بر $۱۰^۹ + ۷$ بگویید .

پاسخ : ابتدا باید ثابت کنیم که اگر c در جهان موجود باشد، آنگاه طول بزرگترین دنباله ای شرط را دارد برابر $۲^c - ۱$ است. برای اثبات استقرا میزنیم.

- **فرض استقرا** : اگر c در جهان موجود باشد، آنگاه طول بزرگترین دنباله ای شرط را دارد برابر $۲^c - ۱$ است.
- **پایه** : حالت $c = ۱$ است که بررسی حکم بدیهی است.
- **گام** : یک دنباله که شرط را دارد، در نظر میگیریم. چون که این دنباله شرط را دارد حتما یک رنگ وجود دارد(رنگ x) که دقیقا یک بار در کل دنباله آمده است. حال دنباله بدون آن رنگ به حداکثر دو بخش تبدیل میشود که خوب بودن این بخش ها، خوب بودن کل دنباله را نتیجه می دهد. چون که هر بازه یا کامل در یکی از بخش ها آمده یا حتما توپ با رنگ x را دارد که یک بار در آن آمده است. چون که رنگ x نمی تواند در آن بخش ها بیاید، از حداکثر $c - ۱$ رنگ می توانیم در هر کدام از بخش ها استفاده کنیم. طبق فرض استقرا، طول هر کدام از بخش ها حداکثر $۲^{c-۱} - ۱$ است که نتیجه میدهد طول دنباله حداکثر

$$\underbrace{۲^{c-۱}-۱}_{\text{بخش اول}}+\underbrace{۲^{c-۱}-۱}_{\text{بخش دوم}}+\underbrace{۱}_{\text{توپ با رنگ } x} = ۲^c - ۱$$

است.

حال برای در اوردن جواب مسئله طبق استدلال هایی که داشتیم یه رنگ هست که دقیقا یک بار در دنباله آمده و دقیقا عضو وسط دنباله است. حال $f(x)$ را جواب مسئله به ازای حالت $n = ۲^x - ۱$ و $k = x$ در نظر میگیریم. حال میگوییم برای محاسبه $f(x)$ ، رنگ توپ وسط x حالت و هر کدام از دو بخش ایجاد شده $f(x-۱)$ حالت دارد. همچنین میدانیم که $f(۱) = ۱$ است. پس

$$f(x) = \begin{cases} ۱ & x = ۱ \\ f(x) \times f(x) \times x & x > ۱ \end{cases}$$

حال جواب مسئله $f(۲۰۰۰۰۰)$ است که باقی مانده بر $۱۰^۹+۷$ برابر ۱۰۵۵۷۶۷۲۸ است.

کد $C++$ سوال :

```
#include <bits/stdc++.h>
using namespace std;
const int Mod = 1e9 + 7;
int main(){
	int k = 200000;
	long long ans = 1;
	for(int i = 2; i <= k; i++) ans = (ans * ans % Mod) * i % Mod;
	cout << ans << '\n';
	return 0;
}
```

اعداد ۰ تا n روی تخته نوشته شده اند. هر مرحله میتوانیم یکی از اعداد که میانگین دوتا از اعداد روی تخته برابر آن است را از روی تخته پاک کنیم. $f(n)$ را کمترین تعداد ممکن اعداد باقی‌مانده به ازای n تعریف میکنیم. برای مثال $f(۱) = ۲$ و $f(۲) = ۲$ است. جمع $f(n)$ به ازای n از ۰ تا ۱۰۰۰ برابر کدام گزینه است؟

پاسخ :

۱) به استقرا روی n ثابت میکنیم $f(n = ۲^k) = ۲$:

$f(n = ۲^{۰} = ۱) = ۲, n = ۲^{k-۱} \rightarrow n = ۲^k$

میتوانیم در $\frac{n}{۲}$ گام، همه اعداد فرد را حذف کنیم (زیرا دو عدد قبل و بعدش که زوجند، حذف نشده اند و این عدد میانگین این دو عدد است). پس بعد از $\frac{n}{۲}$ گام از مجموعه $\{۰, ۱, ۲, \ldots, n\}$ به مجموعه $\{۰, ۲, ۴, \ldots, n\}$ میرسیم. از آنجایی که همه اعداد زوج هستند بدون از دست دادن کلیت مسئله میتوانیم همه اعداد را تقسیم بر ۲ کنیم. در این صورت به مجموعه $\{۰, ۱, ۲, \ldots, \frac{n}{۲}\}$ میرسیم که طبق فرض f آن برابر ۲ است.

۲) ثابت میکنیم $f(n \neq ۲^k) = ۳$:

ابتدا اثبات میکنیم میتوان به ۳ عدد $\{۰, ۱, n\}$ رسید :

در این بخش برعکس سوال عمل میکنیم و مجموعه را از آخر به اول میسازیم (منظور از برعکس این است که میانگین دو عدد پای تخته را به اعداد تخته اضافه کنیم). ابتدا اثبات میکنیم با کمتر از ۳ عدد نمیشود:

لم ۱: دو عدد ۰ و n هیچ‌گاه حذف نمیشوند.

اثبات : بدیهی است.

پس اگر بتوان در آخر به دو عدد رسید، میتوانیم از $\{۰, n\}$ به $\{۰, \ldots, n\}$ برسیم. اثبات میکنیم در این صورت هیچ‌گاه عدد ۱ نوشته نمیشود. ثابت میکنیم در هرمرحله تمام اعداد پای تخته به فرم $a\frac{n}{۲^k}$ هستند. در این صورت اثبات میشود عدد ۱ نمیتواند پای تخته نوشته شود:

$$0 = 0\frac{n}{2^0}, n = 1\frac{n}{2^0}$$

$$\frac{a\frac{n}{2^x} + b\frac{n}{2^y}}{2} = \frac{(a2^y + b2^x)\frac{n}{2^{x+y}}}{2} = (a2^y + b2^x)\frac{n}{2^{x+y+1}}$$

حال اثبات میکنیم میتوان از ۳ عدد $\{۰, ۱, n\}$ به همه اعداد رسید:

$f(۳) = ۳, x < n \to n$

با استفاده از اعداد $\{۰, ۱, n\}$ میتوانیم عدد $\lceil \frac{n}{۲} \rceil$ را اضافه کنیم. سپس طبق فرض همه اعداد ۰ تا $\lceil \frac{n}{۲} \rceil$ ساخته می‌شوند و به کمک $\{\lceil \frac{n}{۲} \rceil - ۱, \lceil \frac{n}{۲} \rceil, n\}$ بقیه اعداد نیز طبق فرض ساخته میشوند.

حال برای محاسبه مقدار خواسته شده میدانیم $f(۰) = ۱$ ، $f(۲^k) = ۲$ که تعداد توان های ۲ تا عدد هزار برابر ۱۰ تا و f بقیه اعداد برابر ۳ است.

$$۱ + ۱۰ \times ۲ + ۹۹۰ \times ۳ = ۲۹۹۱$$

در سرزمین شازززز یک ماز وجود دارد که به صورت جدولی ۱۰۰۰×۱۰۰۰ است که بعضی خانه های آن به جز خانه شروع $(۱, ۱)$ و پایان (n, n) مسدود هستند و نمی توان به آنها رفت. خانه شروع را بالا چپ جدول و خانه پایان را پایین راست جدول در نظر بگیرید. کسی که در یکی از خانه های ماز است در هر مرحله می تواند یا یک خانه به راست برود یا یک خانه به پایین برود (در صورت مسدود نبودن آن خانه).
همچنین می دانیم خانهx, y مسدود است اگر و تنها اگر $x, y \equiv ۰ \pmod{۳}$
شایان و بزرگترین دشمنش در خانه شروع قرار دارند. در هر مرحله هر کدام از آنها یک خانه به راست یا پایین می رود و در نهایت هر دو به خانه پایان می رسند. اگر این دو نفر همزمان به خانه ای به جز خانه شروع و پایان برسند همدیگر را تا حد مرگ کتک می زنند.
حالا وظیفه شما شمردن تعداد مسیر های ممکن این دو نفر (به پیمانه $۱۰^۹ + ۷$) است به صورتیکه هر دو زنده به خانه پایان برسند.

پاسخ : فرض کنید ابتدا شایان به سمت راست می رود و دشمنش به سمت پایین می رود. (در آخر جواب به دست آمده را در ۲ ضرب می کنیم)

$$A = (۱, ۲), B = (۲, ۱), C = (۹۹۹, ۱۰۰۰), D = (۱۰۰۰, ۹۹۹)$$

پس شایان باید از نقطه A به C برود و دشمن باید از B به D برود و سوال تعداد راه زوج مرتب های مسیر شایان و دشمن را می خواهد به طوریکه مسیر ها اشتراک نداشته باشند.
حالا $f(a, b, c, d)$ را تعریف می کنیم تعداد زوج مسیر های شایان و دشمنش به صورتیکه شایان از a به c برود و دشمن از b به d برود. (شرط اشتراک نداشتن را لحاظ نمی کنیم) ادعا می کنیم جواب مسئله برابر است با:
$f(A, B, C, D) - f(A, B, D, C)$

برای اثبات هر زوج مسیر که با هم اشتراک دارند را به یکی از $f(A, B, D, C)$ ها تناظر یک به یک می دهیم. تناظر به این صورت است که اولین محل برخورد را در نظر بگیرید حالا ادامه مسیر شایان را دشمن طی کند و ادامه مسیر دشمن را شایان.
۱) تابع تناظر هر حالتی که مسیر ها اشتراک دارند را به یکی از $f(A, B, D, C)$ ها تناظر می دهد.
۲) توجه کنید که اگر شایان از A به D برود و دشمنش از B به C برود قطعا تقاطعی خواهند داشت. پس می توان اولین محل برخورد را در نظر گرفت و مقصد ها را برعکس کرد. پس تابع تناظر وارون پذیر است.
از ۱ و ۲ نتیجه می گیریم که تناظر یک به یک است.
هر کدام از $f(a, b, c, d)$ ها را هم می توان با استفاده از dp دوبعدی حل کرد. حال جواب مسئله ۸۱۸۸۰۳۴۳ بدست می آید. کد سوال در صفحه بعدی قابل مشاهده است.

کد $C++$ سوال :

```
#include<bits/stdc++.h>
using namespace std;

const int maxn = 1010, mod = 1e9 + 7;
int dp[maxn][maxn];
int calc(int a, int b, int c, int d){
        memset(dp, 0, sizeof dp);
        dp[a][b] = 1;
        for(int i = 1; i < maxn; i++)
            for(int j = 1; j < maxn; j++)
                if(i % 3 != 0 || j % 3 != 0)
                                dp[i][j] += (dp[i - 1][j] + dp[i][j - 1]) % mod;
        return dp[c][d];
}

int main(){
        int A = 1ll * calc(1, 2, 999, 1000) * calc(2, 1, 1000, 999) % mod;
        int B = 1ll * calc(1, 2, 1000, 999) * calc(2, 1, 999, 1000) % mod;
        int ans = (A - B + mod) % mod;
        ans = 2ll * ans % mod;
        if(ans < 0) ans += mod;
        cout << ans << endl;
        return 0;
}
//if you are reading this, send message in Telegram to @the_amoo
//first person will win 5000 Tomans!
```

یک دسته کارت شامل $2n$ کارت که روی آن‌ها عددهای $0, 1, \dots, 2n-1$ نوشته شده است، داده شده است. می‌توانیم با انجام عمل زیر روی این دسته کارت، یک دسته کارت دیگر که در آن ترتیب قرار گرفتن کارت‌ها تغییر کرده است، بسازیم:

ابتدا دسته کارت را به دو دسته که اولی شامل n کارت اول و دومی شامل n کارت باقی‌مانده است، تقسیم می‌کنیم. سپس به ترتیب یک کارت از دسته‌ی اول و یک کارت از دسته دوم برمی‌داریم و این کار را انقدر تکرار می‌کنیم تا تمام کارت‌ها برداشته شوند.

به عنوان مثال اگر شماره‌ی کارت‌ها به ترتیب ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ باشد، پس از انجام عمل فوق، ترتیب قرار گرفتن کارت‌ها به صورت ۱ ۵ ۲ ۶ ۳ ۷ ۴ ۸ خواهد بود.

عمل فوق را بر زدن دسته کارت می‌نامیم.

می‌توان ثابت کرد که برای هر n، اگه دسته کارت را بُر بزنیم، سپس دسته کارت حاصل را دوباره بُر بزنیم و همین کار را تکرار کنیم، بالاخره پس از مدتی به همان دست کارت اولیه می‌رسیم(به عهده خواننده)

برای $n = 4096$ پس از چند بار بر زدن به دسته کارت اولیه می‌رسیم؟

پاسخ :

لم الف : اگر جایگاه کارت ها را از ۰ تا $2n-1$ شماره گذاری کنیم، به ازای هر جایگاه i ، اگر $0 \leq i \leq n-1$ باشد آنگاه کارت جایگاه iام پس از برخوردن در جایگاه $2i$ قرار میگیرد و اگر $n \leq i \leq 2n-1$ باشد آنگاه کارت جایگاه i ام، در جایگاه $2(i-n)+1$ قرار میگیرد.چون j امین کارت دسته ی اول در j امین جایگاه زوج و j امین کارت دسته ی دوم در j امین جایگاه فرد قرار میگیرد.

با توجه به قضیه لم الف ثابت میکنیم به ازای هر $n = 2^k$ و به ازای هر جایگاه مانند x که $(0 \leq x \leq 2^{k+1}-1)$ بعد از $k+1$ بار برزدن کارتی که در بتدا در جایگاه x بوده به جای اولش باز میگردد.

نمایش عدد x در مبنای ۲ را در نظر بگیرید. چون $n = ۲^k$ پس اگر این عدد در مبنای ۲ دارای t رقم باشد، آنقدر در ۲ ضرب میشود تا تعداد ارقامش برابر $k + ۱$ شود. در واقع $k + ۱ - t$ بار در ۲ ضرب میشود. پس به انتهای آن $k + ۱ - t$ صفر اضافه میشود.

حال در اینجا عدد x بزرگتر مساوی $۲k$ میشود و جزء دسته ی دوم کارت ها قرار میگیرد. با توجه به قضیه لم الف اگر جایگاه i در دسته ی دوم قرار داشت کارت موجود در آن به جایگاه $۲(i - n) + ۱$ میرود. میتوان دید برای $n = ۲^k$ این عمل اینگونه تعریف میشود که ابتدا رقم یکمان را از ابتدا بر میداریم (در عملیات $i - n$. زیرا رقم $k + ۱$ ام عدد i یک است) سپس یک رقم صفر به انتهای آن اضافه میشود (وقتی در ۲ ضرب میکنیم) و بعد آن را با یک جمع میکنیم. یعنی درواقع رقم صفری که به انتهای آن اضافه کردیم به یک تبدیل میشود. پس مانند آن است که رقم یک را از ابتدای آن برمیداریم و به انتهای آن اضافه میکنیم.

پس به عبارتی اگر در ابتدای کار ما به ابتدای عدد x ، $k + ۱ - t$ تا رقم صفر اضافه میکنیم، هر عمل ضربدر دو کردن مانند آن است که یک صفر از ابتدای آن برمیداریم و انتهای آن میگذاریم و وقتی در دسته دوم است (در واقع رقم$k + ۱$ ام ۱ است) رقم یک را از ابتدای آن برداشته و به انتهای آن می افزاییم. پس بعد از$k + ۱$ بار به ازای هر جایگاه به خود آن برمیگردیم و حکم ثابت میشود.

برای اثبات مینیمم بودن $k + ۱$ ، عدد در جایگاه دوم دقیقا بعد $۶k + ۱$ به جای خود برمگیردد. ۶ طبق گفته های بالا جواب برای ۴۰۹۶ یا همان $۲^{۱۲}$ برابر ۱۳ است.

زیبایی یک عدد را تعداد ۳ تایی های غیر مرتب از اعداد طبیعی تعریف میکنیم که:

۱- بتوان با انها مثلث قائم الزاویه ساخت.

۲- جمعشان برابر با آن عدد باشد.

تعداد اعداد کمتر مساوی ۱۰۰۰۰۰۰ که زیباییشان ۱ باشد را بیابید.

پاسخ : یک آرایه arr تعریف می کنیم که در نهایت خانه i ام آن باید تعداد مثلث های قائم الزاویه با محیط i باشد. جواب سوال تعداد i هایی است که $arr_i = 1$ است.

ابتدا راه پیچیدگی زمانی $\mathcal{O}(n^2)$ را بررسی می کنیم :

کافیست به ازای هر $a, b \leq n$ که $c = \sqrt{a^2 + b^2}$ عددی طبیعی است arr_{a+b+c} را یکی زیاد کنیم.

قضیه : به ازای هر $x^2 + y^2 = z^2$ اعداد طبیعی r, s, d به صورت یکتایی وجود دارند که :

۱) زوجیت r, s متفاوت است.

۲) اعداد r, s نسبت به هم اولند.

۳) $r > s$

۴) $x = d(2rs), y = d(r^2 - s^2)$ یا $x = d(r^2 - s^2), y = d(2rs)$

۵) $z = d(r^2 + s^2)$

برای اثبات به کتاب نظریه اعداد مریم میرزاخانی فصل ۱۶ مراجعه کنید!

حالا راهی با پیچیدگی زمانی $\mathcal{O}(n.log^{۲})$ را بررسی می کنیم:
حالا از آنجایی که هر x, y, z را به r, s, d تبدیل کردیم داریم :

$$x + y + z = ۲rd(r + s)$$

پس کافیست به ازای هر حالتی از r, d, s که $۲rd(r+s) \leq n$ است $arr_{۲rd(r+s)}$ را یکی زیاد کنیم. توجه کنید که d تنها یک ضریب است پس می توانیم یک آرایه $arr2$ نگه داریم و $arr2_{2r(r+s)}$ را یکی زیاد کنیم.
سپس در نهایت قرار دهیم :

$$arr_i = \sum_{j|i} arr2_j$$

حالا از آنجایی که $۲rd(r + s) \leq n$ پس $s \leq \frac{n}{r}$ پس تعداد r, s های ممکن $\mathcal{O}(n.log)$ است.

کد C++ سوال :

```
#include<bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 10;

int arr[maxn], arr2[maxn];

int solve(int n){
	for(int r = 1; r <= n; r++)
		for(int s = 1; s < r && 2ll * r * (r + s) <= n; s++)
			if(r % 2 != s % 2 && __gcd(r, s) == 1)
				arr2[2 * r * (r+s)] ++;

	for(int i = 1; i <= n; i++)
		for(int j = i; j <= n; j += i)
			arr[j] += arr2[i];

	int ans = 0;
	for(int i = 1; i <= n; i++)
		if(arr[i] == 1) ans ++;

	return ans;
}

int main(){
	int n = 100000;
	cout << solve(n) << endl;
	return 0;
}
```