Nama : Fajar Satria

NRP : 05111940000083

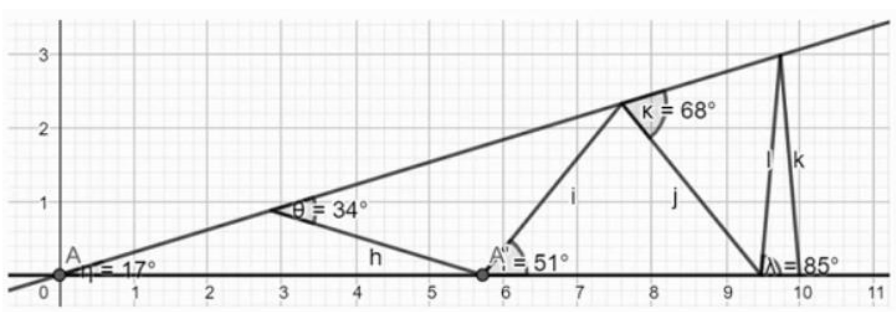
Kelas : B

Resume Kuliah PAA – 26 April 2021

Kemampuan observasi kreatif sangat diperlukan untuk menyelesaikan sebuah persoalan. Misalnya untuk persoalan divide and conquer menyelesaikan persoalan rekursi dengan timelimit yang sangat kecil. Tentu hal tersebut memerlukan banyak sekali observasi dan dilanjutkan dengan pemikiran kreatif untuk menyelesaikan dengan tepat dan cepat persoalan seperti itu.

a. Spiderman

Perkuliahan hari ini dimulai dengan pemanasan asah otak dari persoalan ini. Persoalan ini dapat diselesaikan dengan mencoba-coba menggambarkan beberapa testcase lain.



Sebagai contoh untuk sudut A=17. Lompatan maksimalnya adalah 6 dengan membentuk 5 segitiga sama kaki dengan sudut 17, 34, 51, 68, 85 derajat. Sudut yang dapat dibuat tidak boleh lebih dari 90 derajat.

Untuk sudut A=14. Lompatan maksimalnya adalah 7 dengan membentuk 6 segitiga sama kaki dengan sudut 14, 28, 42, 56, 70, 84 derajat. Dengan begitu dapat disimpulkan bahwa sudut yang dapat dibuat adalah (A, 2A, 3A, … nA) dan lompatan maksimal yang dapat dibuat adalah n+1.

Dikarenakan akurasi yang dibutuhkan sangat tinggi yaitu **10^{-6}** (**A** < **180**) maka penyelesaian akhirnya adalah **floor(90/A-x)** dimana x adalah nilai presisi yang digunakan yakni **10^{-6}** . Atau dapat menggunakan solusi lain yakni **ceil(90/A).**

PSEUDOCODE

SOLVE(A)
1. ans = ceil(90/A)
2. return ans

SOURCECODE

```
#include<cstdio>
#include<cmath>
int main(){
    double a;
    scanf("%lf",&a);
    int res=(ceil)(90.0/a);
    printf("%d\n",res);
    return 0;
}
```

SUBMISSION

#	Submit date	Lang	Time	CPU	Memory	State
8916114	Apr 26, 2021, 9:40:05 AM	C++ 17 (gnu 10.2)	1 ms	1 ms	69	✓ Accepted

b. OAE

Persoalan ini terlihat rumit, dikarenakan harus mencari kombinasi tiap digit dengan ketentuan digit 0 harus berjumlah genap. Dan kombinasi tersebut harus dilanjutkan dengan modulo 314159. Salah satu titik kritis disini adalah jika digit 0 nya berjumlah 0, itu masuk kedalam hitungan atau tidak.

Bilangan 0 itu tidak negatif dan tidak positif, tidak ganjil dan tidak genap, 0 adalah null atau kehampaan. Namun pada perhitungan himpunan 0 tetap dihitung. Atau pada persoalan ini 0 dianggap genap dan tetapi dihitung.

Persoalan ini merupakan permasalahan divide and conquer, pendekatannya pun juga divide and conquer, namun solusi akhirnya merupakan incremental. Karena persoalan ini akan terasa susah untuk dicari solusi terbaik dengan metode kombinatorik.

Analisis yang dapat di gunakan untuk soal ini adalah dengan membuat relasi rekuren yang benar terlebih dahulu dengan menambahkan sebuah terminate case. Dalam kasus ini terminate case nya adalah ketika panjang digit 1 atau a1 = 9 yakni terdapat 9 kemungkinan yang memenuhi. Untuk a2 maka permasalahannya akan terbagi menjadi 3 kemungkinan. Kemungkinan pertama ketika digit a1 adalah valid dan ditambah dengan digit ‘0’ maka solusi a2 tersebut berubah menjadi tidak valid. Kemungkinan kedua ketika digit a1 adalah valid dan ditambah dengan 9 digit yang valid juga {1,2,3,4,5,6,7,8,9} maka solusi a2 tersebut akan tetap valid. Kemungkinan ketiga ketika digit a1 adalah tidak valid dan ditambah dengan digit ‘0’ maka solusi tersebut akan berubah menjadi valid. Maka untuk a_n dapat dituliskan menjadi rumus seperti berikut.

$$\begin{aligned} a_n &= 9a_{n-1} + (10^{n-1} - a_{n-1}) \\ &= 8a_{n-1} + 10^{n-1} \end{aligned}$$

Solusi diatas masih merupakan rekursi jadi ketika a_n dipanggil maka akan memanggil a_{n-1}sampai n=1. Dan akan bermasalah jika n yang dipanggil adalah bilangan yang cukup besar misalnya 10000. Oleh karena itu perlu dilakukan pengoptimalan solusi dengan mengubahnya menjadi solusi incremental.

Dari range $a_1$sampai a_n

$$\begin{aligned} a_n &= 8a_{n-1} + 10^{n-1} \\ a_{n-1} &= 8a_{n-2} + 10^{n-2} \\ &. \\ &. \\ a_2 &= 8a_1 + 10^1 \\ a_1 &= 9 \end{aligned}$$

Dapat dijadikan 1 menjadi

$$a_n = 8^{n-1}a_1 + 8^{n-2}10^1 + 8^{n-3}10^2 + \cdots + +8^0 10^{n-1}$$

Atau

$$\begin{aligned} r &= 8^{n-2}10^1 + 8^{n-3}10^2 + \cdots + 8^0 10^{n-1} \\ 8r &= 8^{n-1}10^1 + 8^{n-2}10^2 + \cdots + 8^0 10^{n-1} \\ 8r &= 8^{n-1}10^1 + 10(8^{n-2}10^1 + 8^{n-3}10^2 + \cdots + 8^1 10^{n-2} + 8^0 10^{n-2}) - 10^n \\ 8r &= 8^{n-1}10^1 + 10r - 10^n \\ 2r &= 10^n - 8^{n-1}10 \\ r &= \frac{1}{2}(10^n - 8^{n-1}10) \end{aligned}$$

Sehingga

$$a_n = 8^{n-1}a_1 + 8^{n-2}10^1 + 8^{n-3}10^2 + \cdots + +8^0 10^{n-1}$$

$$a_n = 8^{n-1}9 + \frac{1}{2}(10^n - 8^{n-1}10)$$
$$a_n = \frac{1}{2}(8^{n-1}8 + 10^n)$$
$$a_n = \frac{1}{2}(8^n + 10^n)$$

Setelah selesai dengan solusi tersebut masih terdapat permasalahan lain yaitu hasil yang diperoleh harus dimodulo dengan 314159. Angka tersebut merupakan bilangan prima. Hal ini bisa diselesaikan dengan **Modular Exponentiation**.

EXAMPLE 12 Use Algorithm 5 to find 3^{644} **mod** 645.

Solution: Algorithm 5 initially sets $x = 1$ and $power = 3$ **mod** $645 = 3$. In the computation of 3^{644} **mod** 645, this algorithm determines 3^{2^j} **mod** 645 for $j = 1, 2, \ldots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the jth position in the binary expansion of 644, which is $(1010000100)_2$), it multiplies the current value of x by 3^{2^j} **mod** 645 and reduces the result modulo 645. Here are the steps used:

$i = 0$: Because $a_0 = 0$, we have $x = 1$ and $power = 3^2$ **mod** $645 = 9$ **mod** $645 = 9$;
$i = 1$: Because $a_1 = 0$, we have $x = 1$ and $power = 9^2$ **mod** $645 = 81$ **mod** $645 = 81$;
$i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81$ **mod** $645 = 81$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 3$: Because $a_3 = 0$, we have $x = 81$ and $power = 111^2$ **mod** $645 = 12{,}321$ **mod** $645 = 66$;
$i = 4$: Because $a_4 = 0$, we have $x = 81$ and $power = 66^2$ **mod** $645 = 4356$ **mod** $645 = 486$;
$i = 5$: Because $a_5 = 0$, we have $x = 81$ and $power = 486^2$ **mod** $645 = 236{,}196$ **mod** $645 = 126$;
$i = 6$: Because $a_6 = 0$, we have $x = 81$ and $power = 126^2$ **mod** $645 = 15{,}876$ **mod** $645 = 396$;
$i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396)$ **mod** $645 = 471$ and $power = 396^2$ **mod** $645 = 156{,}816$ **mod** $645 = 81$;
$i = 8$: Because $a_8 = 0$, we have $x = 471$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111)$ **mod** $645 = 36$.

This shows that following the steps of Algorithm 5 produces the result 3^{644} **mod** $645 = 36$.

Penyelesaian akhir nya adalah dengan menggabungkan kedua solusi diatas.

PSEUDOCODE

MODEXP(a,b,c)

1. r = 1
2. while b>0
3. if (b&1) = 1
4. r = (r * a) mod m
5. b = b >> 1
6. a = (a * a) mod m
7. return r

SOLVE(N)

1. MOD = 314159
2. res = ((MODEXP (8, N, MOD) + MODEXP (10, N, MOD)) * inv2) % MOD
3. return res

SOURCECODE

```
#include<cstdio>
using namespace std;
typedef unsigned long long ll;
const ll MOD = 314159;
ll inv2;
ll modexp(ll b, ll e, ll m){
    ll r = 1;
```

```c
    while(e>0){
        if((e&1)==1)
            r = (r*b) % m;
        e >>= 1;
        b = (b*b) % m;
    }
    return r;
}
int main(){
    ll T,N,hasil;
    inv2 = modexp(2,MOD-2,MOD);
    scanf("%lld",&T);
    while(T--){
        scanf("%lld",&N);
        hasil = ((modexp(8,N,MOD)+modexp(10,N,MOD))*inv2)%MOD;
        printf("%lld\n",hasil);
    }
    return 0;
}
```

SUBMISSION

Fajar: submissions
OAE

ID		DATE	PROBLEM	RESULT	TIME	MEM	LANG
27784824	☐	2021-04-26 10:09:30	OAE	**accepted** edit ideone it	0.01	5.4M	C