

Class ONE | 4/March/2020
Binary search

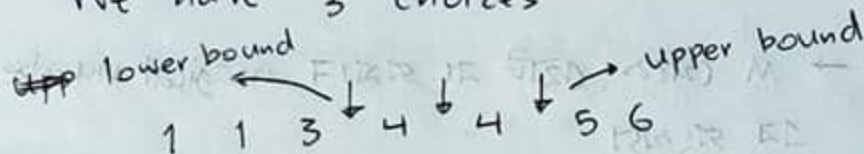
- 1) sorted array
- 2) lower/upper bound

sorted array 1 1 3 4 4 5 6

given value = 4

We have to insert given value in the array, যে array টি sorted থাকে।

We have 3 choices



- 3) Given data গুলো continuous হতে হবে



যদি data গুলো একসাথে থাকে তাহলে Binary search apply করা যাবে।

- 4) Ugly Number → যে নম্বর গুলো 2 বা 3 দ্বারা বিভাজ্য

যখন reverse/inverse কিছু করার সুযোগ থাকে তখন binary search বেশ কাজে লাগতে পারে।

$f(N) \rightarrow N^{\text{th}}$ ugly number

← Nth ugly number

$g(N) \rightarrow t \rightarrow N$ (এক ছোট কয়টা ugly number আছে)

$g(7) \rightarrow 4$

- 2
 - 3
 - 4
 - 6
 - 8
- } ৪ টা

Lecture

৪৭৪. Nth Magical Number :

$g(M) \rightarrow M$ (এক ছোট বা সমান magical number এর সংখ্যা)

এই সমস্যাটা
Problem টা solve
করছি।

$N=1$
 $A=2$
 $B=3$

- 2
- 3
- 4
- 6
- 8
- 9
- 10
- 12
- 14
- 15
- 16

$g(11) \rightarrow 7$

সবচেয়ে Magical number

মূল সমস্যা $\rightarrow f(7) \rightarrow 10$

Reverse Problem $\rightarrow g(10) \rightarrow 7$

We approach
this way

এই এক ছোট কয়টা
magical number
আছে

$g()$ এর definition :

N এর গুণক A এর গুণিতক $\left\lfloor \frac{N}{A} \right\rfloor$
 N " " B এর " $\left\lfloor \frac{N}{B} \right\rfloor$

কমপ্লিমেন্ট
দিয়ে হবে

০০

$$N \text{ এর মধ্যে } \text{lcm}(A, B) \text{ এর গুনিতক } \left\lfloor \frac{N}{\text{lcm}(A, B)} \right\rfloor$$

$$n(A \cup B) = n(A) + n(B) - n(A \cap B) \text{ সূত্র অনুসারে}$$

$$A \quad M = 15$$

$$A = 2$$

$$B = 3$$

২ A এর গুনিতক

2 4 (6) 8 10 (12) 14

B এর গুনিতক

3 (6) 9 (12) 15

$$\text{lcm}(A, B) = 2$$

[6 3 12]

$$g(M) = \left\lfloor \frac{M}{A} \right\rfloor + \left\lfloor \frac{M}{B} \right\rfloor - \left\lfloor \frac{M}{\text{lcm}(A, B)} \right\rfloor$$

$$= 7 + 5 - 2 = 10$$

2 3 4 6 8 9 10 12 14 15 16

$$g(M) = 10$$

$$g(15) = 10$$


```

#define ll long long
class Solution {
public:
    ll g (ll M, int A, int B)
    {
        ll lcm = (A / __gcd(A, B)) * B;
        ll res = (M/A) + (M/B) - (M/lcm);
        return res;
    }

    int nthMagicalNumber (int N, int A, int B)
    {
        ll L, R, M;
        L = 1; // starts here
        R = 1e18; // Large enough number
        int cnt = 70; // for safety
        while (cnt-- && L < R)
        {
            M = L + (R - L) / 2; //  $\frac{L+R}{2} \equiv L + \frac{R-L}{2}$ 
            ll c = g(M) g(M, A, B); // for safety
            if (c < N) // * overflow
                L = M + 1;
            else
                R = M - 1;
        }
        return M % 10000000007;
    }
};

```

Simulation:

০৫

A = 2

B = 4

1 2 3 4 5 6 7 8 9 10 11 12 13 14

2 4 6 8 10 12 14

Magical Numbers

$$g(12) = 6$$

$$g(13) = 6$$

ডিন্স ডিন্স M এর জন্য একই উত্তর
আপডে পায়ে, তাই হাজরতি বামে চান
যায়, চাপডে হবে। তাই lower
bound সিডে হবে।

তাই code এর শেষ নাইনে
Mid return করবেন, lower
bound return করতে হবে।

Homework: lower bound এর code implementation দেখতে হবে।

codeforces 448D

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36

Fig: ONE

i * j দ্বারা পূরণ
করতে হবে

সবগুলো মন্তব্য vector এ
নিনো, দিয়ে sort করতে
হবে। $vc[k-1]$
দিয়ে Kth smallest
number টা প্রিন্ট
করতে হবে।

Binary Search এর range = $[L, R)$

L inclusive | তাই R কে include
R exclusive | করতে হবে R+1 নিতে হবে

১) (১২) → ১ ২ ৩ → ১২ এর চেয়ে ছোট বা সমান সংখ্যা কতগুলো আছে

$$1 \leq N, M \leq 2 \times 10^5$$

Brute force → $O(NM)$

Good enough → $O(N \log M)$

Proposed → $O(N \log(NM))$

Simulation:

First row → 1 2 3 4 5 6 ← size of array
Fig: one $M=5$ || $M=20$ || তাহলে,
 $g(M)=5$ || $g(M)=6$ || $g(M)=\min(M, m)$

One dimensional হলে, $g(M)=M$ হতো [যদি ১ দিয়ে শুরু হয়]

Second row → 2 4 6 8 10 12
Fig: one
২য় row কে ২ দিয়ে ভাগ দিলে
first row পাওয়া।

Generalized function for all rows:

$$g(M) = \min\left(\frac{M}{i}, m\right)$$

একটা table
↑
এই

$$g(35) = g(34) = g(33) = g(32) = g(31) = g(30)$$

তাই আমরাও সাথে চাপতে হবে। সূচকঃ lower bound.

Binary Search on interactive array

CODE :

```
int main ()
```

```
{
```

```
    // n, m, k;
```

```
    cin >> n >> m >> k;
```

```
    // L = 1;
```

```
    // R = (n * m) + 1;
```

```
    // M;
```

```
    int cnt = 70;
```

```
    while (cnt-- && L < R)
```

```
    {
```

```
        M = L + (R - L) / 2;
```

```
        // c = g (M); // O(n * 70)
```

```
        if (c < k)
```

```
            L = M + 1;
```

```
        else
```

```
            R = M;
```

```
    }
```

```
    cout << L << "\n";
```

```
    return 0;
```

```
}
```

```
#include <bits/stdc++.h>
#define ll long long
using namespace std;
ll n, m, k;
```

```
ll g (ll M, int m)
```

```
{
```

```
    ll cnt = 0;
```

```
    for (int i = 1; i <= n; i++)
```

```
    { cnt += min(M / i, m);
```

```
    }
```

```
    return cnt;
```

```
}
```

g() is complexity $O(n)$.

09