YTP Problem 3：陰晴圓缺 - 解答 / Abundance Sum - Solution

劉至軒

Thursday 12th August, 2021

解答

令 $N = R - L + 1$ 為區間長度。

Subtask 1 - 3/15

這個就是暴力解，只需要將 $[L, R]$ 的每一個數字掃過暴力枚舉因數即可。複雜度 $O(N\sqrt{N})$，拿三分。不過因為是基本分數，$R = 10^4$ 也不是非常大，所以 $O(N^2)$ 也是可以過的哦！

Subtask 2 - 8/15

我們可以假設 $L = 1$，因為題目其實就只是叫你求兩個前綴的差而已。這樣子，就可以列式，所求的為

$$\begin{aligned}\sum_{x=1}^{N} \Delta(x) &= \sum_{x=1}^{N} \left(x - \sum_{y<x,\, y \mid x} y \right) \\ &= \sum_{x=1}^{N} x - \sum_{\substack{y<x \\ y \mid x}} y \\ &= \frac{N(N+1)}{2} - \sum_{x=1}^{N} \sum_{\substack{y<x \\ y \mid x}} y\end{aligned}$$

後面的雙重求和比較麻煩，但是其實只需要換一個角度看就很簡單了——與其看 x 求因數，不如看 y 求倍數：對於一個 y，它會被幾個 x 數到？其實就是 **除了他以**

外的倍數在 N 以內有幾個那麼多次。也就是說，

$$
\begin{aligned}
&\frac{N(N+1)}{2}-\sum_{x=1}^{N}\sum_{\substack{y<x\\ y\,|\,x}}y\\
=&\frac{N(N+1)}{2}-\sum_{y=1}^{N}\sum_{\substack{y<x\\ y\,|\,x\\ x\in[1,N]}}y\\
=&\frac{N(N+1)}{2}-\sum_{y=1}^{N}\left(\lfloor\frac{N}{y}-1\rfloor\right)y\\
=&N(N+1)-\sum_{y=1}^{N}\lfloor\frac{N}{y}\rfloor y
\end{aligned}
$$

所以對於每一個 y 都計算一次就好了，複雜度 $O(N)$。

Subtask 3 - 15/15

我們繼續沿用前面的式子：因為 $\lfloor\frac{N}{y}\rfloor$ 的值只有 $O(\sqrt{N})$ 種（事實上，不會超過 $2\sqrt{N}$ 種），所以用數論分塊可以將 $\lfloor\frac{N}{y}\rfloor$ 相同的一起算，然後就只是 $\lfloor\frac{N}{y}\rfloor$ 乘上一些的連續數字相乘，總複雜度 $O(\sqrt{N})$。

Solution

Let $N = R - L + 1$ be the length of the given interval.

Subtask 1 - 3/15

All that needs to be done is for every number within range, perform a search for all its proper divisors. This solution takes $O(N\sqrt{N})$ time.

Subtask 2 - 5/15

W.L.O.G. let $L = 1$, since all we are looking for is the difference of two prefix sums. Then what we are looking for is

$$\begin{aligned}\sum_{x=1}^{N} \Delta(x) &= \sum_{x=1}^{N} \left(x - \sum_{y<x,\, y \mid x} y \right) \\ &= \sum_{x=1}^{N} x - \sum_{\substack{y<x \\ y \mid x}} y \\ &= \frac{N(N+1)}{2} - \sum_{x=1}^{N} \sum_{\substack{y<x \\ y \mid x}} y\end{aligned}$$

What's annoying is the double summation at the end. Fortunately, all that's required is a change of perspective: if we not think about divisors but instead of multiples, we can switch the order of summations and sum by y instead. Then for every y, it will be counted however many proper multiples under N it has times. So the summation becomes:

$$\begin{aligned}&\frac{N(N+1)}{2} - \sum_{x=1}^{N} \sum_{\substack{y<x \\ y \mid x}} y \\ =&\frac{N(N+1)}{2} - \sum_{y=1}^{N} \sum_{\substack{y<x \\ y \mid x \\ x\in[1,N]}} y \\ =&\frac{N(N+1)}{2} - \sum_{y=1}^{N} \left(\lfloor \frac{N}{y} - 1 \rfloor \right) y \\ =&N(N+1) - \sum_{y=1}^{N} \lfloor \frac{N}{y} \rfloor y\end{aligned}$$

And so we can just iterate over all y for a linear solution (i.e. runs in $O(N)$).

Subtask 3 - 15/15

All that's needed to get full marks in this problem is just to realise that there are only $O(\sqrt{N})$ different values (actually, no more than $2\sqrt{N}$ distinct values) that $\lfloor \frac{N}{y} \rfloor$ can take. So we just need to group them up by $\lfloor \frac{N}{y} \rfloor$ and multiply it by the sum of the integers in that range. This solution runs in $O(\sqrt{N})$ time.

官解 / AC Code

```cpp
#include <iostream>
#define int long long int
using namespace std;
const int M = 1e9 + 7, I2 = 5e8 + 4;

inline int mult(int a, int b){
    return ((a % M) * (b % M)) % M;
}

inline int isum(int l, int r){ //returns \sum_{k=l}^{r} k mod M
    int res = (mult(r, r + 1) - mult(l, l - 1) + M) % M;
    return res * I2 % M;
}

inline int sum(int x){
    if(x <= 1) return x;
    int s = mult(x, x + 1);
    for(int i = 1; i <= x; i++){
        //for all t \in [i, \lfloor x / \lfloor x/i \rfloor \rfloor], \lfloor x/t \rfloor is the same.
        int l = i;
        int r = (x / (x / i));
        s = (s - mult(x / l, isum(l, r)) + M) % M;
        i = r;

    }
    return s;
}

signed main() {
    int L, R;
    cin >> L >> R;
    cout << (sum(R) - sum(L - 1) + M) % M << endl;
}
```