팩토리얼

키-파

March 31, 2021

1 동기 및 해야 할 일

주어진 자연수 n을 10진법으로 나타내었을 때 맨 앞의 두 자리를 구하는 쉬운 방법은 *Stirling series*를 이용하는 것입니다.

$$\log n! = \left(n + \frac{1}{2}\right) \log n - n + \frac{1}{2} \log 2\pi + \sum_{i=1}^{\infty} \frac{B_{2i}}{2i(2i-1)n^{2i-1}} \tag{1}$$

여기서 B_{2i}는 $2i$-번째 *Bernoulli number*입니다. 이 식은

$$|B_{2n}| \sim 4\sqrt{\pi n} \left(\frac{n}{\pi e}\right)^{2n} \tag{2}$$

이기에 수렴하지 않으나, 식 (1)의 부분합

$$S_m(n) := \left(n + \frac{1}{2}\right) \log n - n + \frac{1}{2} \log 2\pi + \sum_{i=1}^{m} \frac{B_{2i}}{2i(2i-1)n^{2i-1}}$$

의 오차 범위

$$|S_{m-1}(n) - \log n!| \leq \frac{B_{2m}}{2m(2m-1)n^{2m-1}} =: \epsilon_m(n)$$

가 알려져 있으며, 이때 (2)를 이용해

$$\epsilon_n(n) \approx \sqrt{\frac{\pi}{n}} (\pi e)^{-2n}$$

임을 알 수 있습니다. $\pi^2 e^2 \approx 72.9271$임을 참고하면, $n \geq 10^4$일 때 최소 10^{-10^4}의 정밀도로 $\log_{10} n!$을 계산할 수 있음은 분명합니다.

그러나 만일 $582469933139689265! \approx 62.999999999942 * 10^{10094774197006386079}$처럼 앞 두 자리만 남겼을 때 정수와 매우 가까운 수를 실제로 찾을 수 있다면 이야기가 달라집니다. 이 문제가 문제로서 성립하기 위해서는, 모든 $n \leq N$에 대해 $\log_{10} n!$의 가수부가 $\log_{10} 1$, $\log_{10} 1.1$, $\cdots$, $\log_{10} 9.9$, $\log_{10} 10$ 중 어느 것과도 K보다는 멀리 떨어져 있다는 것을 증명해야 합니다.

그래서 고민 끝에, 이런 상황을 모두 문제 상황으로 제시해 놓고 앞 두 자리만 남겼을 때 정수에 가깝도록 하는 수 n을 범위 내에서 찾아서 찍으라는 문제를 냈습니다. 우리는 "두 자리 정수에 가까운 팩토리얼 찾기" 문제를 풀어야 합니다.

2 modulo minimum

먼저 다음과 같은 문제를 생각합니다.

> 주어진 a, b, $e \geq 0$, N에 대해 $\min_{0 \leq x \leq e}(ax + b \mod N)$을 효율적으로 구하는 방법은?

일반성을 잃지 않고 $\gcd(a, N) = 1$이라고 가정할 수 있습니다. 이 문제를 입력에 대한 다항 시간에 효율적으로 풀 수 있습니다.

1. 만일 $2a \geq N$이면 $(a, b) := (N - a, ae + b \mod N)$으로 놓으면 동등한 문제를 푸는 것이기 때문에,[1] 항상 $a \leq \frac{N}{2}$이라 가정할 수 있습니다.

2. (Base Case) 만일 e가 충분히 작으면 (예를 들어 $e \leq 10$) 전부 다 해 볼 수 있습니다.

3. (Base Case) 만일 $ae + b < N$이면 답은 b입니다.

4. (Recursive Case) 그렇지 않으면, b가 답의 후보가 될 수 있습니다.
   - $ae + b \geq N$이기 때문에 a씩 더해 가면서 한 번은 N을 넘길 수 있습니다.
   - 이 수 $s := b + a \cdot \lfloor \frac{N-b+a-1}{a} \rfloor - N$를 생각합니다.
   - s부터는 $-N \mod a$씩만 더해가면서 보면 되는데, 이는 a씩 더하면 modulo a에 대한 불변량이 생기는데 한 번 올라갈 때마다 N씩 빠지기 때문입니다.
   - 이렇게 더해 가다 만일 a를 넘어가는 경우, a를 넘지 않을 때까지 a를 빼 주어야 합니다.
   - 이렇게 해서 N을 넘을 수 있는 횟수는 자명하게 $e_{\text{next}} = \lfloor \frac{ae+b}{N} \rfloor - 1$입니다.

   따라서 답은 b와 $(a, b, N, e) := (-N \mod a, s, a, e_{\text{next}})$에 대해 푼 경우 중 작은 쪽이 됩니다.

이를 이용해서, 주어진 a, b, s, e, N에 대해 $ax + b \mod N \in [s, e]$가 되는 가장 작은 음이 아닌 정수 x를 효율적으로 구할 수 있습니다. 마찬가지로 일반성을 잃지 않고 $\gcd(a, N) = 1$이라고 가정합니다.[2]

1. (이 경우에서 고려할 필요는 없지만) 만일 $a = 0$이면 $s \leq b \leq e$인 경우 $x = 0$이 최소가 되고, 아닌 경우는 답이 없습니다.

2. $y \in [s, e]$에 대해서 $ax + b = y \mod N$이 되는 x의 값을 구할 수 있습니다: $x = a^{-1}(y - b) \mod N$. 이 값의 최솟값을 구하면 됩니다.

3. $(a, b, e, N) = \left(a^{-1} \mod N, \quad (s - b) \cdot a^{-1} \mod N, \quad e - s, \quad N\right)$에 대해서 modmin 문제를 푼 값이 답이 됩니다.

3 오차 범위가 제한된 근사

원래 문제로 돌아갑시다. 이 문제를 빠르게 푸는 기본적인 아이디어는 어떤 n에서 적은 수의 연산만 사용해서 가능한 한 높이 "점프"하는 것입니다.

범위 $[n, n + k]$의 $\log_{10}$ 값을 전부 $\log_{10}(n + c)$로 근사한다고 합시다: 이때 $c = \lceil \frac{k+1}{2} \rceil$입니다. 그러면 $0 \leq i \leq k$에 대해 $\log_{10}(n + i)! \approx \log_{10} n! + i \log_{10}(n + c)$로 근사할 수 있습니다. 아래는 첫 번째 근사입니다.

$$
\begin{aligned}
\log_{10}(n+i)! - (\log_{10} n! + i \log_{10}(n+c)) &= \frac{1}{\log 10} \sum_{j=1}^{i} \log\left(1 + \frac{j-c}{n+c}\right) \\
&\leq \frac{1}{\log 10} \sum_{j=1}^{i} \frac{j-c}{n+c} \\
&= \frac{1}{(n+c)\log 10} \cdot \left(\frac{i(i+1)}{2} - ci\right) \\
&= \frac{i}{(n+c)\log 10} \cdot \left(\frac{i+1}{2} - c\right) \leq 0.
\end{aligned}
$$

[1] $ax + b$라는 식을 $a(e - x) + b$로 고친 것입니다.
[2] 이 경우 s와 e를 재조정하는 것이 약간 tricky할 수 있습니다. 가능한 s와 e로 재조정한 이후 $s > e$일 수 있는데, 이 경우는 답이 없습니다.

$$
\begin{aligned}
\frac{1}{\log 10}\sum_{j=1}^{i}\log\left(1+\frac{j-c}{n+c}\right) &\geq \frac{1}{\log 10}\sum_{j=1}^{i}\left(\frac{j-c}{n+c}-\frac{(j-c)^2}{2(n+c)^2}\right)\\
&\geq \frac{1}{\log 10}\sum_{j=1}^{i}\frac{j-c}{n+c}-\frac{c^2 i}{2(n+c)^2\log 10}\\
&\geq \frac{i}{(n+c)\log 10}\left(\frac{i+1}{2}-c\right)-\frac{c^3}{(n+c)^2\log 10}\\
&\geq \frac{-(4c^2-4c+1)}{8(n+c)\log 10}-\frac{c^3}{(n+c)^2\log 10}\\
&\geq -\left(\frac{c^2-c+1}{2(n+c)\log 10}+\frac{c^3}{(n+c)^2\log 10}\right).
\end{aligned}
$$

그러나 컴퓨터는 실제로 유리수밖에 계산할 수 없기 때문에, $\log_{10} n!$이 A로, $\log_{10}(n+c)$가 B로 한 번 더 근사되었을 것입니다. 우리는 $A \leq \log_{10} n! \leq A+p$, $B \leq \log_{10}(n+c) \leq B+p$라고 가정합니다. 그러면 두 번째 근사를 시행할 수 있습니다.

$$0 \leq (\log_{10} n! + i\log_{10}(n+c)) - (A+iB) = p + ip \leq (k+1)p.$$

A와 B의 가수부를 정수로 정확히 남겼다고 합시다. 이 값들은 $a := AN \mod N$, $b := BN \mod N$이 될 것입니다. 그러면 $a+bi \mod N$은 $\log(N+i)!$의 가수부의 근사가 됩니다. 우리는 이 값이 (근사 오차를 포함하여) 적당한 범위 안에 있는 모든 n을 찾아내는 것에 집중합니다. $(a+bi)$는 참값보다 최대 $\epsilon_{\text{lower}} := \lceil (k+1)pN \rceil$만큼 작을 수 있고,

$$\epsilon_{\text{upper}} := \left\lceil \left(\frac{c^2-c+1}{2(n+c)\log 10}+\frac{c^3}{(n+c)^2}\right)N \right\rceil$$

만큼 클 수 있으므로, 구하고자 하는 범위가 $[-\varepsilon, \varepsilon]$이라면 $(a+bi)$로 찾을 때는 참값 범위의 lower bound에는 $\lceil \varepsilon N \rceil + \epsilon_{\text{lower}}$만큼 빼 줘야 하고, upper bound에는 $\lceil \varepsilon N \rceil + \epsilon_{\text{upper}}$만큼 더해 줘야 합니다.

4 parameter searching

- p 값은 가능한 한 커야 속도 면에서 우위를 점할 수 있습니다. 그러나 p 값이 너무 커지면 연산에 필요한 정밀도를 만족시킬 수 없습니다. 위에서 $pN \leq 1$이어야 함은 자명하나, $pN = 1$인 데 논리상 아무런 문제가 없으므로 $N = 1/p$로 그냥 두면 됩니다.

- k 값은 너무 크면 작은 곳에서의 허용 오차 범위가 커집니다. 작은 곳에서는 허용 오차가 클 필요가 없는데 큰 오차로 계산하니 값을 너무 작은 곳에서 찾아 버려서, jump가 실제로 k만큼 일어날 수 없습니다. k가 너무 작으면 애초에 jump할 수 있는 범위가 작아집니다.

- $k \sim C \cdot n^{1/3}$으로 두는 것이 좋다는 추측이 있는데, 전혀 터무니없는 추측은 아니고 Benford's law에 기반을 두고 있습니다. 실제로 상수를 잘 잡으면 이 경우 반반 정도로 jump를 시행할 수 있어서, $\mathcal{O}(n^{2/3}P(\log n))$ 시간에 문제를 해결할 수 있습니다.

대회 시간 안에 만점을 받으려면 구간을 잘 나눠서 멀티 코어로 돌리거나 GPU 가속을 시행하는 등의 어려움이 있습니다.