# 如何暴力地求和

$n + e$

*Tsinghua University*

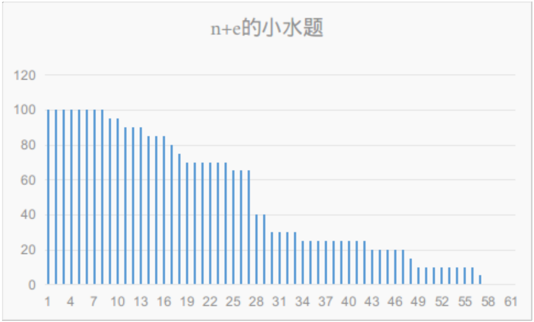
2016 年 12 月 5 日

- 有一个多项式函数 $f(x)$，最高次幂为 $x^m$，定义变换 $Q$:

$$Q(f) = \sum_{k=0}^{n} f(k) \binom{n}{k} x^k (1-x)^{n-k}$$

- 现在给定函数 $f$ 和 $n, x$，求 $Q(f) \bmod 998244353$。

- 100 分：8 人（昨天半个场 AC 实在是太可怕了）
- 95 分：2 人（计划通）
- 90 分：3 人（计划通）
- 85 分：3 人（计划通）
- 65 分以上：27 人
- 平均分：47.7



- 通过尝试可以发现答案是一个关于 $n$ 和 $x$ 的多项式，且最高次项为 $m$。题目里面第二段已经给出相应的提示。
- 比如，当多项式函数为 $f: x \mapsto x^2$ 时，所要求的是

$$Q(x^2) = \sum_{k=0}^{n} k^2 \binom{n}{k} x^k (1-x)^{n-k}$$

- 如果推公式/找规律能力够强的话，这个表达式的答案是

$$n^2x^2 - nx^2 + nx$$

- 此为 7 号点答案。

- 一种想法是，利用乘法分配律，将函数 $f$ 的每一个项单独考虑，最后直接相加即可。
- 问题转化为，化简表达式

$$Q(x^c) = \sum_{k=0}^{n} k^c \binom{n}{k} x^k (1-x)^{n-k}$$

其中 $c \in \{1, 2, \cdots, m\}$。

- 不好化简？
- $Q(f)$ 有什么性质？

- 对于 10% 的数据，$n, m \leq 1000$ 且 $n = m$

- 对于 10% 的数据，$n, m \le 1000$ 且 $n = m$
- 考察 for 语句的使用、组合数的求法、快速幂的实现。

1. 试题大意
2. 得分情况
3. 解题思路
4. 算法介绍
   - 算法一
   - 算法二
   - 算法三
   - 算法四
   - 算法五
   - 算法六？

- 对于 20% 的数据，$n \leq 100000$

- 对于 20% 的数据，$n \leq 100000$
- 加上一个多项式插值就好了。剩下不变。
- 也许常数好的话可以拿 25 分？反正都开 O2 了
- 然而实际情况却是……？
- 有人拿部分分了。计划通！

- 对于 5% 的数据，$m = 1$

- 对于 5% 的数据，$m = 1$
- 函数 $f: x \mapsto a + bx$，分开考虑。
- $Q(f) = Q(a + bx) = a \cdot Q(1) + b \cdot Q(x)$
- 由二项式定理，$Q(1) = 1$。
- 由题目第二段，$Q(x) = nx$。
- 所以 $Q(f) = a + bnx$。

# 为啥 $Q(x) = nx$？

$$
\begin{aligned}
& \sum_{k=0}^{n} k\binom{n}{k}x^k(1-x)^{n-k} \\
= \quad & \sum_{k=0}^{n} k\frac{n!}{k!(n-k)!}x^k(1-x)^{n-k} \\
= \quad & \sum_{k=0}^{n} \frac{n(n-1)!}{(k-1)!(n-k)!}x^k(1-x)^{n-k} \\
= \quad & n\sum_{k=0}^{n} \binom{n-1}{k-1}x^k(1-x)^{n-k} \\
= \quad & nx\sum_{k=0}^{n} \binom{n-1}{k-1}x^{k-1}(1-x)^{n-k}
\end{aligned}
$$

# 为啥 $Q(x) = nx$？

- 的确有“母函数求导”这种方法可以搞，因为我第一次做的时候就用那个方法。
- 我真的忘记了我是怎么做的了 qaq

- 对于 10% 的数据，$m = 2$
- 对于 5% 的数据，$m = 3$

- 对于 10% 的数据，$m = 2$
- 对于 5% 的数据，$m = 3$
- 暴力找规律，$m = 2$ 的答案在前面。
- 不找规律怎么做呢？

- $Q(x^2)$ 怎么求？

- $Q(x^2)$ 怎么求？我真不会直接搞
- $Q(x^2-x)$ 怎么求？

- $Q(x^2)$ 怎么求？我真不会直接搞
- $Q(x^2-x)$ 怎么求？这个很好求啊～ $x^2-x=x(x-1)=x^{\underline{2}}$
- 仿照刚才的思路，把组合数拆开，下降幂塞进去就好了。
- 你看出题人多良心，题目塞满了提示，塞满了部分分。(然而各位大爷们似乎一个个都直接艹过去了)
- 身败名裂

$$
\begin{aligned}
& \sum_{k=0}^{n} k^2 \binom{n}{k} x^k (1-x)^{n-k} \\
= \ & \sum_{k=0}^{n} n(n-1) \binom{n-2}{k-2} x^k (1-x)^{n-k} \\
= \ & n^2 x^2 \sum_{k=0}^{n} \binom{n-2}{k-2} x^{k-2} (1-x)^{n-k} \\
= \ & n^2 x^2 \cdot Q(1) = n^2 x^2
\end{aligned}
$$

同理有

$$
Q(x^c) = n^c x^c
$$

- 现在问题转化为：将多项式函数 $f$ 表示成下降幂的形式。即求出 $d_0, d_1, \cdots, d_m, s.t.$

$$\sum_{k=0}^{m} c_k x^k = \sum_{k=0}^{m} d_k x^{\underline{k}}$$

- 假设求完了 $d_k$，那么统计答案的时候花 $O(m)$ 的时间扫一遍就好了。
- 其实 $c_k$ 也不是给好的。题目中给出了 $f(0..m)$ 具体的值。

- 定义差分运算：$\triangle f(x) = f(x+1) - f(x)$，可以理解为离散求导。
- 由组合数的定义

$$\triangle\left(\binom{x}{k}\right) = \binom{x}{k-1}$$

- 设 $b_k = k!d_k$，则

$$f(x) = b_m\binom{x}{m} + b_{m-1}\binom{x}{m-1} + \cdots + b_1\binom{x}{1} + b_0\binom{x}{0}$$

$$\triangle f(x) = b_m\binom{x}{m-1} + b_{m-1}\binom{x}{m-2} + \cdots + b_1\binom{x}{0}$$

$$\triangle f(0) = b_1$$

$$\triangle^2 f(x) = b_m\binom{x}{m-2} + b_{m-1}\binom{x}{m-3} + \cdots + b_2\binom{x}{0}$$

$$\triangle^2 f(0) = b_2$$

$$\vdots$$

$$\triangle^k f(0) = b_k$$

$$c_k = k!\triangle^k f(0)$$

# 举个栗子

- $f(x) = x^3$
- 

| $x$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $\triangle^0 f$ | 0 | 1 | 8 | 27 |
| $\triangle^1 f$ | 1 | 7 | 19 | |
| $\triangle^2 f$ | 6 | 12 | | |
| $\triangle^3 f$ | 6 | | | |

$$x^3 = 6\binom{x}{3} + 6\binom{x}{2} + 1\binom{x}{1} + 0\binom{x}{0}$$

- 验题人是用斯特林数做的。由于做法太过奇怪所以就被我强行卡掉了
- m 其实是可以更大的。至少 10w 没问题。
- 就是加一个 FFT 优化咯。->(Ask 汪乐平/叶珈宁/袁宇韬/徐泽涛/丁力煌/吕欣)（6/8 计划通）
- 考虑到这是第一题所以就没加咯。
- 出题人比较懒就只写了平方算法……（听说有人平方被卡？怪我咯）
- myy 15min 怒 A orz

$$
\begin{aligned}
d_k &= \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} a_i \\
&= \sum_{i=0}^{k} (-1)^{k-i} a_i \frac{k!}{i!(k-i)!} \\
\frac{d_k}{k!} &= \sum_{i=0}^{k} \frac{(-1)^{k-i}}{(k-i)!} \frac{a_i}{i!}
\end{aligned}
$$

## 算法六？

```
int n, m, x;
int f[maxm + 5];

int d[maxm + 5];

int main()
{
#ifdef matthew99
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif
	scanf("%d%d%d", &n, &m, &x);
	REP(i, 0, m + 1) scanf("%d", f + i);
	int ans = 0;
	int now = 1;
	REP(i, 0, m + 1)
	{
		(ans += (LL)f[0] * now % Mod) %= Mod;
		REP(j, 0, m - i) (f[j] = f[j + 1] - f[j]) %= Mod;
		now = (LL)now * x % Mod;
		now = (LL)now * (n - i) % Mod;
		now = (LL)now * fpm(i + 1, Mod - 2, Mod) % Mod;
	}
	(ans += Mod) %= Mod;
	printf("%d\n", ans);
	return 0;
}
```

- 达成成就：在清华集训出过题 ~
- Thanks for listening!