遗忘的集合 (set) 解题报告

Tangjz

2017 年 5 月 18 日

题目大意

- 有一个正整数集合 S，每个元素都不大于 n，注意集合可以为空集，但不会有重复元素，n 不一定在 S 中
- 定义 $f(x)$ 表示有多少种方法可以把 x 用 S 中一些元素之和表示，注意一个表示里可以出现多个相同的元素，两个表示不同当且仅当存在某个元素的出现次数不同
- 根据 $f(1), f(2), \cdots, f(n) \pmod p$ 还原出字典序最小的 S
- 假设 S 大小为 m，S 中的数字按升序排列为 $s_1, s_2, \cdots, s_m$
- 对于 100% 的数据，有 $1 \le n < 2^{18}$, $10^6 \le p < 2^{30}$, p 是质数

震惊! 数据范围竟然是

测试点编号	n	p	特殊约定
1	$n = 5$	$p = 1000003$	无特殊约定
2	$n \leq 20$	同最大限制	
3	$n \leq 25$		
4			
5	$n \leq 5000$		$s_m \leq 40$
6			
7	$n \leq 8000$	$p = 1000003$	无特殊约定
8		$p = 1000000007$	
9		同最大限制	
10			$m = s_m$
11	同最大限制		
12			
13			
14			
15		$p = 998244353$	无特殊约定
16		$p = 991668907$	
17		$p = 1000000007$	
18		同最大限制	
19			
20			

整体情况

- 100分：待定
- 80分：待定
- 70分：待定
- 50分：人数待定
- 25分：人数待定
- 低于25分：人数待定
- 平均分：99.99
- 中位数：100

一些吐槽

- FFT 考不考啊，不考不学了啊？

一些吐槽

- FFT 考不考啊，不考不学了啊？

  这位选手请开始你的表演 

一些吐槽

- FFT 考不考啊，不考不学了啊？

  这位选手请开始你的表演 

- 说好的送温暖呢？

一些吐槽

- FFT 考不考啊，不考不学了啊？

  这位选手请开始你的表演 

- 说好的送温暖呢？

  搜索 25 分，背包 50 到 70 分，FFT 100 分 

一些吐槽

- FFT 考不考啊，不考不学了啊？

  这位选手请开始你的表演 

- 说好的送温暖呢？

  搜索 25 分，背包 50 到 70 分，FFT 100 分 

- 为什么不写 Special Judge ？

一些吐槽

- FFT 考不考啊，不考不学了啊？

  这位选手请开始你的表演 

- 说好的送温暖呢？

  搜索 25 分，背包 50 到 70 分，FFT 100 分 

- 为什么不写 Special Judge ？

  解是唯一的，出题人不忍心写假的 SPJ 忽悠你 

做法 1

- Case 1: $n = 5, p = 1000003$ (不是样例)

做法 1

- Case 1: $n = 5, p = 1000003$ (不是样例)
- 看懂题就能拿到了
- 时间复杂度 $\mathcal{O}(1)$，期望得分 5 分

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$
- 按字典序枚举每种可能的答案，把每个元素 x 看作是完全背包的一种体积为 x 的物品，计算出对应的 f 与输入比对

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$
- 按字典序枚举每种可能的答案，把每个元素 x 看作是完全背包的一种体积为 x 的物品，计算出对应的 f 与输入比对
- 时间复杂度 $\mathcal{O}(2^n n^2)$ ，期望得分 10 分

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$
- 按字典序枚举每种可能的答案，把每个元素 x 看作是完全背包的一种体积为 x 的物品，计算出对应的 f 与输入比对
- 时间复杂度 $\mathcal{O}(2^n n^2)$ ，期望得分 10 分
- 字典序可以用 dfs 序表示，做法改成 dfs ，边 dfs 边转移

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$
- 按字典序枚举每种可能的答案，把每个元素 x 看作是完全背包的一种体积为 x 的物品，计算出对应的 f 与输入比对
- 时间复杂度 $\mathcal{O}(2^n n^2)$ ，期望得分 10 分
- 字典序可以用 dfs 序表示，做法改成 dfs ，边 dfs 边转移
- 时间复杂度 $\mathcal{O}(2^n n)$ ，期望得分 25 分

做法 2

- Case 1, 2: $n \leq 20$, Case 3, 4: $n \leq 25$
- 按字典序枚举每种可能的答案，把每个元素 x 看作是完全背包的一种体积为 x 的物品，计算出对应的 f 与输入比对
- 时间复杂度 $\mathcal{O}(2^n n^2)$ ，期望得分 10 分
- 字典序可以用 dfs 序表示，做法改成 dfs ，边 dfs 边转移
- 时间复杂度 $\mathcal{O}(2^n n)$ ，期望得分 25 分
- 多出的 5 分来自 Case 10 ，没有回溯，复杂度变为 $\mathcal{O}(n^2)$

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40
- 保证了有解，实际上只用看前 $\min(n, 40)$ 个函数值即可判定

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40
- 保证了有解，实际上只用看前 $\min(n, 40)$ 个函数值即可判定
- 中途相遇法，将 S 集合按权值拆成两部分，分别正向、逆向处理信息，类似双向 bfs

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40
- 保证了有解，实际上只用看前 $\min(n, 40)$ 个函数值即可判定
- 中途相遇法，将 S 集合按权值拆成两部分，分别正向、逆向处理信息，类似双向 bfs
- 空间复杂度 $\mathcal{O}(2^{20}20)$ ，只能开一个大数组，可结合 dfs 实现

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40
- 保证了有解，实际上只用看前 $\min(n, 40)$ 个函数值即可判定
- 中途相遇法，将 S 集合按权值拆成两部分，分别正向、逆向处理信息，类似双向 bfs
- 空间复杂度 $\mathcal{O}(2^{20}20)$ ，只能开一个大数组，可结合 dfs 实现
- 时限较宽，信息很好区分，查找可以排序二分或 hash

做法 3

- Case 1 - 6: $n \leq 5000$, S 的最大值 s_m 不超过 40
- 保证了有解，实际上只用看前 $\min(n, 40)$ 个函数值即可判定
- 中途相遇法，将 S 集合按权值拆成两部分，分别正向、逆向处理信息，类似双向 bfs
- 空间复杂度 $\mathcal{O}(2^{20}20)$ ，只能开一个大数组，可结合 dfs 实现
- 时限较宽，信息很好区分，查找可以排序二分或 hash
- 时间复杂度 $\mathcal{O}(2^{20}20)$ ，期望得分 30 分

做法 4

- Excuse me? 为什么要 dfs ?

做法 4

- Excuse me? 为什么要 dfs ?
- 换个想法，从小到大枚举每个元素 x ，如果当前 $f(x) = 1$ ，则该元素属于 S ，然后把 f 变换成不含 x 时的背包方案数

做法 4

- Excuse me? 为什么要 dfs？解唯一
- 换个想法，从小到大枚举每个元素 x，如果当前 $f(x) = 1$，则该元素属于 S，然后把 f 变换成不含 x 时的背包方案数
- 完(sheng)全(cheng)背(han)包(shu)的可逆性 $f[j] \leftarrow f[j] - f[j - x]$

做法 4

- Excuse me? 为什么要 dfs？解唯一
- 换个想法，从小到大枚举每个元素 x，如果当前 $f(x) = 1$，则该元素属于 S，然后把 f 变换成不含 x 时的背包方案数
- 完(sheng)全(cheng)背(han)包(shu)的可逆性 $f[j] \leftarrow f[j] - f[j - x]$
- 正确性：枚举到 x 时已经没有小于 x 的数会影响现在的 f，若有解则 $f(x)$ 只能为 0 或 1

做法 4

- Excuse me? 为什么要 dfs？解唯一
- 换个想法，从小到大枚举每个元素 x，如果当前 $f(x) = 1$，则该元素属于 S，然后把 f 变换成不含 x 时的背包方案数
- 完(sheng)全(cheng)背(han)包(shu)的可逆性 $f[j] \leftarrow f[j] - f[j - x]$
- 正确性：枚举到 x 时已经没有小于 x 的数会影响现在的 f，若有解则 $f(x)$ 只能为 0 或 1
- Case 1 - 10: $n \leq 8000$

做法 4

- Excuse me? 为什么要 dfs？解唯一
- 换个想法，从小到大枚举每个元素 x，如果当前 $f(x) = 1$，则该元素属于 S，然后把 f 变换成不含 x 时的背包方案数
- 完(sheng)全(cheng)背(han)包(shu)的可逆性 $f[j] \leftarrow f[j] - f[j - x]$
- 正确性：枚举到 x 时已经没有小于 x 的数会影响现在的 f，若有解则 $f(x)$ 只能为 0 或 1
- Case 1 - 10: $n \leq 8000$
- 时间复杂度 $\mathcal{O}(n^2)$，期望得分 50 分

做法 5

- 预计一些选手开始玩泥巴

做法 5

- 预计一些选手开始玩泥巴
- Case 10 - 14: $n < 2^{18}$, 解为前 m 个正整数

做法 5

- 预计一些选手开始玩泥巴
- Case 10 - 14: $n < 2^{18}$, 解为前 m 个正整数
- 特殊约定使得我们只需要求解 m

做法 5

- 预计一些选手开始玩泥巴
- Case 10 - 14: $n < 2^{18}$, 解为前 m 个正整数
- 特殊约定使得我们只需要求解 m
- 对于 $m = k$ 和 $m > k$ 的情况，它们的 $f(k+1)$ 一定不等

做法 5

- 预计一些选手开始玩泥巴
- Case 10 - 14: $n < 2^{18}$, 解为前 m 个正整数
- 特殊约定使得我们只需要求解 m
- 对于 $m = k$ 和 $m > k$ 的情况，它们的 $f(k+1)$ 一定不等
- 求出有多少种划分方式可以将 $1, 2, \cdots, n$ 分为若干个正整数之和 (划分数) 即可

做法 5

- 预计一些选手开始玩泥巴
- Case 10 - 14: $n < 2^{18}$, 解为前 m 个正整数
- 特殊约定使得我们只需要求解 m
- 对于 $m = k$ 和 $m > k$ 的情况，它们的 $f(k+1)$ 一定不等
- 求出有多少种划分方式可以将 $1, 2, \cdots, n$ 分为若干个正整数之和 (划分数) 即可
- 玩泥巴选手：我不会算划分数啊 QAQ

做法 5

- 对于不小于 T 的数字，其在任一划分方案中的出现次数不超过 $\frac{n}{T}$，令 $g[j][k]$ 表示用 j 个不小于 T 的数字，总和是 k 的方案数，按照划分方案的最小值是否是 T 分类讨论，则有

$$g[j][k] = g[j-1][k-T] + g[j][k-j]$$

做法 5

- 对于不小于 T 的数字，其在任一划分方案中的出现次数不超过 $\frac{n}{T}$，令 $g[j][k]$ 表示用 j 个不小于 T 的数字，总和是 k 的方案数，按照划分方案的最小值是否是 T 分类讨论，则有

  $$g[j][k] = g[j-1][k-T] + g[j][k-j]$$

- 之后枚举小于 T 的数字求解完全背包即可，T 取 $\mathcal{O}(\sqrt{n})$ 可得最优时间复杂度 $\mathcal{O}(n\sqrt{n})$，结合做法 4 期望得分 70 分

做法 5

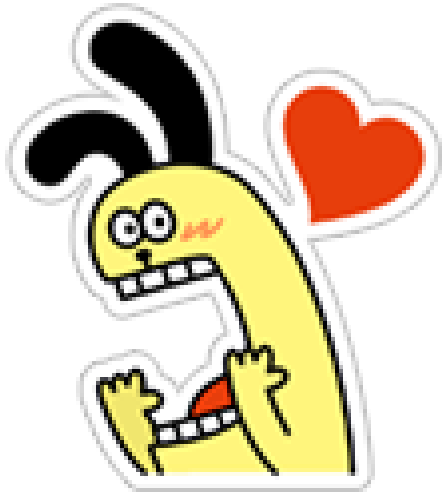
- 对于不小于 T 的数字，其在任一划分方案中的出现次数不超过 $\frac{n}{T}$，令 $g[j][k]$ 表示用 j 个不小于 T 的数字，总和是 k 的方案数，按照划分方案的最小值是否是 T 分类讨论，则有

  $$g[j][k] = g[j-1][k-T] + g[j][k-j]$$

- 之后枚举小于 T 的数字求解完全背包即可，T 取 $\mathcal{O}(\sqrt{n})$ 可得最优时间复杂度 $\mathcal{O}(n\sqrt{n})$，结合做法 4 期望得分 70 分
- 也可用五边形数定理 $\mathcal{O}(n\sqrt{n})$ 计算前 n 个正整数的划分数，参见附录1，注意做法 5 并没有办法处理一般情况

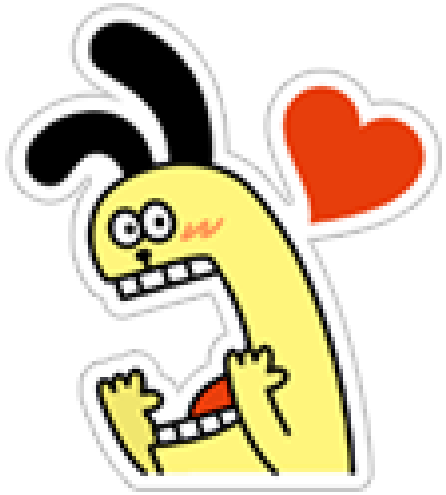
做法 6

- 掉线选手请准备重连，我们要发车了



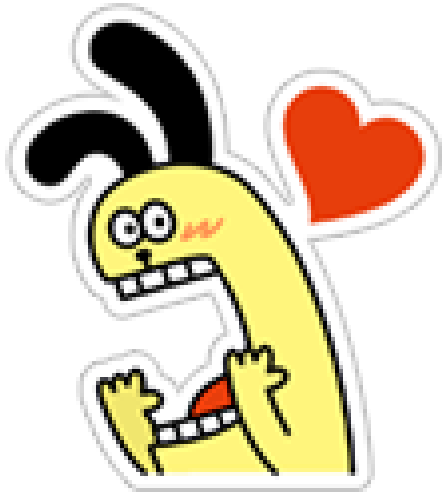
做法 6

- 开始飙车
- 定义生成函数 (形式幂级数) $F(z) = 1 + \sum_{j=1}^{n} f(j) z^j$



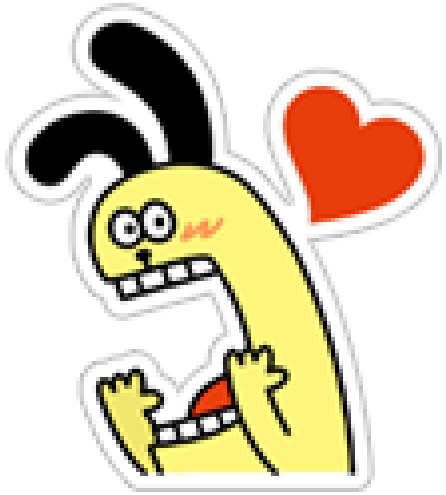
做法 6

- 开始飙车
- 定义生成函数 (形式幂级数) $F(z) = 1 + \sum_{j=1}^{n} f(j) z^j$
- 由题意得 $F(z) \equiv \prod_{j \in S} \frac{1}{1 - z^j} \pmod{z^{n+1}}$



做法 6

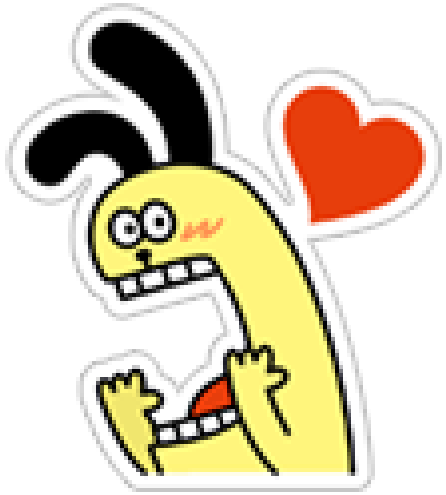
- 开始飙车
- 定义生成函数 (形式幂级数) $F(z) = 1 + \sum_{j=1}^{n} f(j) z^j$
- 由题意得 $F(z) \equiv \prod_{j \in S} \frac{1}{1 - z^j} \pmod{z^{n+1}}$
- 由于 $F(z)$ 的常数项为 1，则 $\ln F(z)$ 在 $z = 0$ 处的级数展开依旧是一个形式幂级数



做法 6

- 开始飙车
- 定义生成函数 (形式幂级数) $F(z) = 1 + \sum_{j=1}^{n} f(j) z^j$
- 由题意得 $F(z) \equiv \prod_{j \in S} \frac{1}{1 - z^j} \pmod{z^{n+1}}$
- 由于 $F(z)$ 的常数项为 1，则 $\ln F(z)$ 在 $z = 0$ 处的级数展开依旧是一个形式幂级数
- $\ln F(z) = \int \frac{\mathrm{d}F}{F} = \int \frac{\mathrm{d}F}{\mathrm{d}z} \frac{\mathrm{d}z}{F} = \int \frac{F'(z)}{F(z)} \mathrm{d}z$

  熟悉多项式初等运算的选手应该并不陌生



做法 6

- 接下来的内容需要用到一个化简技巧 $\ln\left(\frac{1}{1-x}\right)=\sum_{j>0}\frac{x^j}{j}$

做法 6

- 接下来的内容需要用到一个化简技巧 $\ln\left(\dfrac{1}{1-x}\right)=\sum\limits_{j>0}\dfrac{x^j}{j}$
- 原式取对数有 $\int\dfrac{F'}{F}\mathrm{d}z\equiv\sum\limits_{j\in S}\sum\limits_{k>0}\dfrac{z^{jk}}{k}\pmod{z^{n+1}}$

做法 6

- 接下来的内容需要用到一个化简技巧 $\ln\left(\dfrac{1}{1-x}\right)=\sum\limits_{j>0}\dfrac{x^j}{j}$
- 原式取对数有 $\int\dfrac{F'}{F}\mathrm{d}z\equiv\sum\limits_{j\in S}\sum\limits_{k>0}\dfrac{z^{jk}}{k}\pmod{z^{n+1}}$
- 上式两边常数项均为 0，同时求导得 $\dfrac{F'(z)}{F(z)}\equiv\sum\limits_{j\in S}\sum\limits_{k>0}jz^{jk-1}$

做法 6

- 接下来的内容需要用到一个化简技巧 $\ln\left(\dfrac{1}{1-x}\right)=\sum\limits_{j>0}\dfrac{x^j}{j}$
- 原式取对数有 $\int\dfrac{F'}{F}\mathrm{d}z\equiv\sum\limits_{j\in S}\sum\limits_{k>0}\dfrac{z^{jk}}{k}\pmod{z^{n+1}}$
- 上式两边常数项均为 0，同时求导得 $\dfrac{F'(z)}{F(z)}\equiv\sum\limits_{j\in S}\sum\limits_{k>0}jz^{jk-1}$
- 难道要 FFT/NTT ？

精度怎么办… 不是很懂多项式那套理论怎么办…

做法 6

- 接下来的内容需要用到一个化简技巧 $\ln\left(\frac{1}{1-x}\right)=\sum_{j>0}\frac{x^j}{j}$
- 原式取对数有 $\int\frac{F'}{F}\mathrm{d}z\equiv\sum_{j\in S}\sum_{k>0}\frac{z^{jk}}{k}\pmod{z^{n+1}}$
- 上式两边常数项均为 0，同时求导得 $\frac{F'(z)}{F(z)}\equiv\sum_{j\in S}\sum_{k>0}jz^{jk-1}$
- 难道要 FFT/NTT ？

  精度怎么办… 不是很懂多项式那套理论怎么办…
- 右边的形式幂级数可以 $\mathcal{O}(n\log n)$ 进行正向、逆向构造，而 $F(z)$ 显然存在逆元，可以移动到右边

做法 6.1

- 把式子化得漂亮点!

做法 6.1

- 把式子化得漂亮点!
- 定义 $g(x)=\sum\limits_{\substack{d|x\\d\in S}}d$，则有 $g(x)=xf(x)-\sum\limits_{j=1}^{x-1}g(j)f(x-j)$

做法 6.1

- 把式子化得漂亮点!
- 定义 $g(x)=\sum\limits_{\substack{d|x\\d\in S}}d$，则有 $g(x)=xf(x)-\sum\limits_{j=1}^{x-1}g(j)f(x-j)$
- 求出 $g(1),g(2),\cdots,g(n)\pmod p$ 后很容易求出 S

做法 6.1

- 把式子化得漂亮点!
- 定义 $g(x)=\sum\limits_{\substack{d|x\\d\in S}}d$，则有 $g(x)=xf(x)-\sum\limits_{j=1}^{x-1}g(j)f(x-j)$
- 求出 $g(1),g(2),\cdots,g(n)\pmod p$ 后很容易求出 S
- Case 1 - 10: $n\le 8000$
- 时间复杂度 $\mathcal{O}(n^2)$，期望得分 50 分，结合做法 5 期望 70 分

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT

- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT

- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$

- “看到 UOJ 膜数我就默写了一个 NTT 出来”

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT

- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$
- “看到 UOJ 膜数我就默写了一个 NTT 出来”
- Case 16: $n < 2^{18}, p = 991668907 = 3^{13} \cdot 2 \cdot 311 + 1$

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT
- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$
- “看到 UOJ 膜数我就默写了一个 NTT 出来”
- Case 16: $n < 2^{18}, p = 991668907 = 3^{13} \cdot 2 \cdot 311 + 1$
- “听说你会写三进制 NTT ”“三进制 NTT 不是随手写”

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT
- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$
- “看到 UOJ 膜数我就默写了一个 NTT 出来”
- Case 16: $n < 2^{18}, p = 991668907 = 3^{13} \cdot 2 \cdot 311 + 1$
- “听说你会写三进制 NTT ”“三进制 NTT 不是随手写”
- 时间复杂度 $\mathcal{O}(n \log n)$ ，期望得分 5 分 ~~Case 16 居然卡常数~~

做法 6.2

- 好气啊， FFT 精度好像有点问题？

  没关系，我们准备了两个测试点供选手 NTT
- Case 15: $n < 2^{18}, p = 998244353 = 2^{23} \cdot 7 \cdot 17 + 1$
- “看到 UOJ 膜数我就默写了一个 NTT 出来”
- Case 16: $n < 2^{18}, p = 991668907 = 3^{13} \cdot 2 \cdot 311 + 1$
- “听说你会写三进制 NTT ”“三进制 NTT 不是随手写”
- 时间复杂度 $\mathcal{O}(n \log n)$ ，期望得分 5 分 ~~Case 16 居然卡常数~~
- 不希望选手在这两个测试点上花费太多时间，做法见附录2

做法 6.3

- 冷静一下，思考 FFT mod $(10^9 + 7)$ 怎么写

做法 6.3

- 冷静一下，思考 FFT mod (10^9+7) 怎么写
- 将多项式的系数 v 写成二元组 $a \cdot T + b$ 形式，取 $T = \lceil \sqrt{p} \rceil$

  可以使 a, b 尽量小，一个多项式可以写成两个小系数多项式

做法 6.3

- 冷静一下，思考 FFT mod (10^9+7) 怎么写
- 将多项式的系数 v 写成二元组 $a \cdot T + b$ 形式，取 $T = \lceil \sqrt{p} \rceil$ 可以使 a, b 尽量小，一个多项式可以写成两个小系数多项式
- 直接做是 4 次 DFT 与 3 次 IDFT，有些写法会在 $n \geq 2^{15}$ 时出现精度误差，需要注意

做法 6.3

- 冷静一下，思考 FFT mod (10^9+7) 怎么写
- 将多项式的系数 v 写成二元组 $a \cdot T + b$ 形式，取 $T = \lceil \sqrt{p} \rceil$ 可以使 a, b 尽量小，一个多项式可以写成两个小系数多项式
- 直接做是 4 次 DFT 与 3 次 IDFT，有些写法会在 $n \geq 2^{15}$ 时出现精度误差，需要注意
- 分治 FFT $\mathcal{O}(n \log^2 n)$，多项式求逆 $\mathcal{O}(n \log n)$，期望得分 $50 - 100$ 分

做法 6.3

- 冷静一下，思考 FFT mod (10^9+7) 怎么写
- 将多项式的系数 v 写成二元组 $a \cdot T + b$ 形式，取 $T = \lceil \sqrt{p} \rceil$ 可以使 a, b 尽量小，一个多项式可以写成两个小系数多项式
- 直接做是 4 次 DFT 与 3 次 IDFT，有些写法会在 $n \geq 2^{15}$ 时出现精度误差，需要注意
- 分治 FFT $\mathcal{O}(n \log^2 n)$，多项式求逆 $\mathcal{O}(n \log n)$，期望得分 $50 - 100$ 分
- 得分前提是精度足够好，具体做法见附录3和附录4

做法 6.4

- 4 次 DFT 的输入只用到了系数的实部，3 次 IDFT 的结果只有系数的实部有用

做法 6.4

- 4 次 DFT 的输入只用到了系数的实部， 3 次 IDFT 的结果只有系数的实部有用，能否将虚部也用起来？

做法 6.4

- 4 次 DFT 的输入只用到了系数的实部，3 次 IDFT 的结果只有系数的实部有用，能否将虚部也用起来？ 可以
- 两个多项式的点值合并进一个多项式的点值是很简单的；两个多项式的系数分别存于实部与虚部，进行 DFT 后也可以线性分离出来对应的贡献，具体见附录5

做法 6.4

- 4 次 DFT 的输入只用到了系数的实部， 3 次 IDFT 的结果只有系数的实部有用，能否将虚部也用起来？ 可以
- 两个多项式的点值合并进一个多项式的点值是很简单的；两个多项式的系数分别存于实部与虚部，进行 DFT 后也可以线性分离出来对应的贡献，具体见附录5
- 这样是 2 次 DFT 与 2 次 IDFT ，并且精度和效率都有所改善

做法 6.4

- 4 次 DFT 的输入只用到了系数的实部， 3 次 IDFT 的结果只有系数的实部有用，能否将虚部也用起来？ 可以
- 两个多项式的点值合并进一个多项式的点值是很简单的；两个多项式的系数分别存于实部与虚部，进行 DFT 后也可以线性分离出来对应的贡献，具体见附录5
- 这样是 2 次 DFT 与 2 次 IDFT ，并且精度和效率都有所改善
- 套用多项式求逆即可，时间复杂度 $\mathcal{O}(n \log n)$ ，期望 100 分

表演结束

Thank you!

附录 1 - 五边形定理

- $\prod_{j>0} \frac{1}{1-z^j}$ 的 z^n 系数等于 n 划分成偶数个互异正整数的方案数减去 n 划分成奇数个互异正整数的方案数
- 对于 $n \neq \frac{k(3k \pm 1)}{2}$ 的情况，奇数划分方案和偶数划分方案可以形成双射，两两抵消
- 否则恰好存在一种情况无法映射，此时 z^n 的系数等于 $(-1)^k$
- 因此 $\prod_{j>0} \frac{1}{1-z^j} \pmod{z^{n+1}}$ 只有 $\mathcal{O}(\sqrt{n})$ 项系数非 0
- 它的逆元，也即划分数的生成函数，可以 $\mathcal{O}(n\sqrt{n})$ 计算

附录 2 - 三进制 NTT

- 长度为 3^k 的序列进行离散傅里叶变换，并使用缩系中阶为 3^k 的元素代替单位复根
- Split and Merge 过程
  - 一个长度为 3^{k+1} 的序列的系数分裂成三个长度为 3^k 的序列的系数：将原序列中模 3 余 r 的元素依次放进第 r 个序列
  - 三个长度为 3^k 的序列的点值合并成一个长度为 3^{k+1} 的序列的点值：$F(\omega_{3^{k+1}}^{a3^k+b}) = \sum_{0 \le r < 3} (\omega_{3^{k+1}}^{a3^k+b})^r \cdot F_r(\omega_{3^k}^b)$
- Split 和 Merge 可以分别迭代实现
- 时间复杂度 $\mathcal{O}(3n \log_3 n)$

附录 3 - 分治 FFT

- 可以解决一类 $f_n = \sum_{0 \le j \le n} g_j h_{n-j}$ 的卷积，其中 g_j 的取值与 f_k $(0 \le k < j)$ 有关，例如本题 $g(x) = xf(x) - \sum_{j=1}^{x-1} g(j)f(x-j)$
- 考虑当前分治需要求出 $g(x), x \in [L, R]$ (的贡献)
  - 取 $M = \left\lfloor \frac{L+R}{2} \right\rfloor$，首先求解 $[L, M]$ 的情况
  - 利用算好的 $g(x), x \in [L, M]$ 和 f 进行长度不小于 $R - L$ 的卷积，求出它们对 $g(x), x \in [M+1, R]$ 产生的贡献
  - $g(x), x \in [M+1, R]$ 中未产生的贡献一定来自于 $[M+1, R]$，继续求解 $[M+1, R]$ 的情况
- 时间复杂度 $\mathcal{O}(n \sum_{k \ge 0} \log \frac{n}{2^k}) = \mathcal{O}(n \log^2 n)$

附录 4 - 多项式求逆

- 利用牛顿迭代法求解 $F(z)u(z) \equiv 1 \pmod{z^{n+1}}$
  - 定义 $G(u) = F(z)u - 1$ $(u \in R[z])$，则有 $G'(u) = \dfrac{\mathrm{d}G}{\mathrm{d}u} = F(z)$
  - 初始解 $u_0 = 1$，迭代方程 $u_{k+1} = u_k - \dfrac{G(u_k)}{G'(u_k)}$
  - 每次迭代精度加倍，故有 $G(u_k) \equiv 0 \pmod{z^{2^k}}$
  - $G(u_k)$ 在模 $z^{2^{k+1}}$ 意义下是 z^{2^k} 的倍数，迭代方程中 $\dfrac{1}{G'(u_k)}$ 只需要求其模 z^{2^k} 意义下的值，也即 u_k，方程化为 $u_{k+1} = u_k(1 - G(u_k))$
- 时间复杂度 $\mathcal{O}(\sum\limits_{k \geq 0} \dfrac{n}{2^k} \log \dfrac{n}{2^k}) = \mathcal{O}(n \log n)$

附录 5 - 实数序列对的离散傅里叶变换合并

- 定义 $i=\sqrt{-1}$，定义 $a+bi$ 的共轭复数 $(a+bi)^*=a-bi$，定义 $F_u[0], F_u[1], \cdots, F_u[n-1]$ 表示对长度为 n 的多项式 $u(x)$ 进行 DFT 得到的点值序列
- 尝试合并$A(x), B(x)$，构造

$$P(x)=A(x)+iB(x), Q(x)=A(x)-iB(x)$$

- 注意到 ω_n^k 的共轭复数是 ω_n^{n-k}，则有

$$F_P[k]=A(\omega_n^k)+iB(\omega_n^k)=(A(\omega_n^{n-k})-iB(\omega_n^{n-k}))^*=F_Q[n-k]^*$$

- 对 $P(x)$ 进行 DFT 后可还原出 $A(x), B(x)$ 分别 DFT 的结果