E5 H

题目分析

小水獭幼儿园出现了疫情！

小水獭幼儿园可以被抽象为一个数轴，其中共有 n 名学生，第 i 名学生位于坐标 x_i，任意两名学生的坐标均不相同。对于每名学生均有 $\frac{1}{2}$ 的概率被感染，且每名学生是否被感染相互独立。

基于最新的科学研究，如果第 i 名学生和第 j 名学生都被感染（$i < j$），那么需要额外投入 $2^{|x_i - x_j|}$ 的核酸费用。设核酸费用之和的数学期望为 C，容易证明 $4C$ 是一个整数，请你求解 $4C$ 对 $998\,244\,353$ 取模后的结果。

同学们众志成城，根据科学精准防控，一定能战胜疫情！

第一行一个正整数 t（$1 \le t \le 10$），表示数据组数。

对于每组数据，第一行一个正整数 n（$2 \le n \le 10^5$），含义如题目所示。

第二行 n 个正整数 $x_1, x_2, \ldots, x_n$（$1 \le x_i \le 10^5$），含义如题目所示。

保证 $x_i \ne x_j$ 对 $1 \le i < j \le n$ 成立。

先对n名学生排序，去掉绝对值

猜想期望：

考虑一个点对(i, j)，如果某种可能里存在这对点，
那么i，j都存在，P=1/4（i, j是满足两点分布的独立变量）

$$E = \frac{1}{4}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} 2^{x_i - x_j}$$

题目分析

因此我们求4C，即为求： $E=\sum_{\mathrm{i}=0}^{n-1}\sum_{j=0}^{i-1}2^{\mathrm{x^i}-\mathrm{x^j}}$

为了方便求解，我们将其调整为这样的形式： $=\sum_{i=0}^{i-1}2^{x^i}\sum_{j=0}^{i-1}2^{-x^j}$

在每次循环（i 从0-n-1）中，我们存储 $\sum_{j=0}^{i-1}2^{-x^j}$ ，i 每增加1，答案增加 $2^{\mathrm{x^i}}\sum_{j=0}^{i-1}2^{-x^j}$

```
int n, x[100];

int ans = 0, sum = 0;
for (int i = 0; i < n; i++)
{
    int v = pow(2, x[i]); // v 为2的x[i]次方
    ans = (ans + v * sum);
    sum = (sum + 1 / v);
}
```

（左图为以上分析的代码化）

题目分析

我们一直忽略了一个条件!　　输出一行一个非负整数表示 $4C$ 对 998 244 353 取模后的结果。

如果像这样粗暴的加上模，那么会造成以下问题:

```
int n, x[100];

int ans = 0, sum = 0;
for (int i = 0; i < n; i++)
{
    int v = pow(2, x[i]) % MOD; // v 为2的x[i]次方
    ans = (ans + v * sum) % MOD;
    sum = (sum + 1 / v) % MOD;
}
```

1. Pow函数的精度问题
2. Sum运算包含除法，而在取模运算中，我们不喜欢除法。
   除了因为无法拆分造成溢出外，还可能产生精度问题

解决办法:

1.自己写一个求幂函数——快速幂（比较简单，略讲）

2.使用乘法逆元代替除法——费马小定理

```
ll qpow(ll a, ll n)
{
    ll re = 1;
    while (n)
    {
        if (n & 1)
            re = (re * a) % MOD;
        n >>= 1;
        a = (a * a) % MOD;
    }
    return re % MOD;
}
```

快速幂（略）

费马小定理和乘法逆元

- 什么是乘法逆元:
  - x*y = x/(1/y) 1/y就是y的逆元

- 为什么要求乘法逆元:
  - 模运算中除法性质不好，我们想用乘法

- 怎么求乘法逆元:
  - 费马小定理（读者自证）

对于任意整数 a，有 $a^p \equiv a \pmod{p}$。

对于本题，改写为: $a^{p-2} \equiv a^{-1}$ (mod p)

最终版本

```
int n;
scanf("%d", &n);
int x[100005] = {0}; //存放每个点的位置坐标
ll ans = 0, sum = 0;
for (int i = 0; i < n; i++)
    scanf("%d", &x[i]);
sort(x, x + n); //排序取绝对值

//下面是重点

for (int i = 0; i < n; i++)
{
    ll v = qpow(2, x[i]);
    ans = (ans + v * sum) % MOD;
    sum = (sum + qpow(v, MOD - 2)) % MOD;
}
printf("%lld\n", ans);
```

谢谢！