集合Hash

2021-03-02 | 2022-09-06 | OI & ACM | 5
2.3k | 2 分钟

简介

集合 $hash$ 大概就是用某种 $hash$ 方法对集合内的每个元素赋一个值，然后集合的 $hash$ 值为集合中元素的 $hash$ 值用某种方法组合起来

手写hash表

比 unordered_map 还要快一点

```
struct Hashtable {
    static const int N = 200010, p = 20011203; // 200011203 == 750MB
    struct node {
        ll k; int v, next;
    } pool[N]; vector<int> clr; int tab[p], top;

    void init() {
        for (auto t : clr) tab[t] = 0;
        clr.clear(); clr.shrink_to_fit(); top = 0;
    }
    inline int hash(ll k) { return (k % p + p) % p; }
    void update(ll k, int v) {
        int tk = hash(k);
        for (int i = tab[tk]; i; i = pool[i].next) {
            node &u = pool[i];
            if (u.k == k) return u.v = v, void();
        }
        pool[++top] = { k, v, tab[tk] };
        tab[tk] = top; clr.push_back(tk);
    }
    int query(ll k) {
        int tk = hash(k);
        for (int i = tab[tk]; i; i = pool[i].next) {
```

```
            node &u = pool[i];
            if (u.k == k) return u.v;
        } return -1;
    }
} H;
```

生成随机数的方法

```
default_random_engine e(time(0));
uniform_int_distribution<int> u1(0, 9); // 整型
uniform_real_distribution<float> u2(0, 9); // 浮点
```

不可重集

1. 我们一般采用的方法都是给每个随机一个 ull 范围内的整数，然后将这些元素异或起来表示集合的 $hash$ 值
2. 随机一个 g，将元素 v 变成 g^v，然后将这些元素加起来对大质数取模表示集合的 $hash$ 值
3. 首项为 a，差为 d，长度为 n 的等差数列，我们一般取一个较小的数 k 作为指数，这个等差数列的 $hash$ 值为 $\sum_{i=0}^{n-1}(a+di)^k=\sum_{i=0}^{n-1}\sum_{j=0}^{k}\binom{k}{j}a^j(di)^{k-j}=\sum_{j=0}^{j}\binom{k}{j}a^jd^{k-j}\sum_{i=0}^{n-1}i^k$，可以 $O(nk)$ 预处理，$O(k)$ 单次求，一般多取几个 k 就行了

可重集

1. 随机一个 g，将元素 v 变成 g^v，然后将这些元素加起来对大质数取模表示集合的 $hash$ 值

例题

1. 简要题意：给定一张 n 个 m 条边的简单无向连通图，现在有 q 次询问，每次询问给定若干条边，求将这些边的删掉后原图是否连通，询问之间独立，强制在线

   $n\le 10^5,m\le 5\times 10^5$

   简要题解：注意到只有删边，我们考虑对原图建立一棵生成树，首先只有删掉这棵生成树上的边时才有可能导致原图不连通，那如果我们删掉树上的一条边 (u,v) 使原图不连通，则说明所有能够连接 u 和 v 这两个连通块的非树边都被删掉了

那么相当于是每次删掉一个树边后判断是否该树边对应的非树边集合是否已经全部被删到，我们可以为每条非树边随机一个 ull 范围内的整数，同时树边的值为它所对应的非树边的值的异或和，那么每次删边我们需要把边加入线性基，之后只需要判断线性基中是否出现 0 即可

bzoj 3569 DZY Loves Chinese II

2. 简要题意：给定一个长度为 n 的序列 a_i，现在有 m 次询问，每次询问给定 $[l,r]$ 和 d，问 $[l,r]$ 内的数在模 $1e9+7$ 意义下是否可以构成一个差为 d 的等差数列

   $n,m \leq 2\times 10^5$

   简要题解：我们考虑如果不是在模意义下，那么我们可以使用最经典的 $hash$ 方法，随机一个底数 g，数 x 的 $hash$ 值为 g^x，那么对于一个首项为 a，公差为 d，长度为 $n-1$ 的等差数列的 $hash$ 值为 $g^{\sum_{i=0}^{n-1} a+di}$，这个东西是一个等比数列可以直接套用等比数列求和公式，但可惜的是指数位置上不能直接取模，所以我们不能使用这个方法来 $hash$

   我们考虑更换 $hash$ 方法为随机一个指数 k，数 x 的 $hash$ 值为 x^k，那么等差数列的 $hash$ 值为 $\sum_{i=0}^{n-1}(a+di)^k=\sum_{i=0}^{n-1}\sum_{j=0}^{k}\binom{k}{j}a^j(di)^{k-j}=\sum_{j=0}^{j}\binom{k}{j}a^jd^{k-j}\sum_{i=0}^{n-1}i^k$，这个东西可以 $O(nk)$ 预处理，$O(k)$ 单次求

   但是如果我们取的 k 太小的话可能会被卡掉，所以我们需要多取几个，这里我取了 $k\in[1,10]$，时间复杂度为 $O((n+m)k)$，处理的好的话可以避免所有快速幂

   CF 1599F Mars

---------- 本文结束 感谢阅读 ----------

Tech # hash

‹ CF 986C AND Graph

Luogu P6688 可重集 ›