

普通莫队算法

形式

假设 $n=m$ 那么对于序列上的区间询问问题，如果从 $[l,r]$ 的答案能够 $O(1)$ 扩展到 $[l-1,r],[l+1,r],[l,r+1],[l,r-1]$ (即与 $[l,r]$ 相邻的区间) 的答案，那么可以在 $O(n\sqrt{n})$ 的复杂度内求出所有询问的答案。

实现

离线后排序，顺序处理每个询问，暴力从上一个区间的答案转移到下一个区间答案（一步步移动即可）。

排序方法

对于区间 $[l,r]$ 以 l 所在块的编号为第一关键字， r 为第二关键字从小到大排序。

模板

```
inline void move(int pos, int sign) {
    // update nowAns
}

void solve() {
    BLOCK_SIZE = int(ceil(pow(n, 0.5)));
    sort(queries, queries + m);
    for (int i = 0; i < m; ++i) {
        const query &q = queries[i];
        while (l > q.l) move(--l, 1);
        while (r < q.r) move(r++, 1);
        while (l < q.l) move(l++, -1);
        while (r > q.r) move(--r, -1);
        ans[q.id] = nowAns;
    }
}
```

例题 & 代码

例题「国家集训队」小 Z 的袜子

题目大意：

有一个长度为 n 的序列 $\{c_i\}$ 现在给出 m 个询问，每次给出两个数 l,r 从编号在 l 到 r 之间的数中随机选出两个不同的数，求两个数相等的概率。

思路：莫队算法模板题。

对于区间 $[l,r]$ 以 l 所在块的编号为第一关键字 r 为第二关键字从小到大排序。

然后从序列的第一个询问开始计算答案，第一个询问通过直接暴力算出，复杂度为 $O(n)$ 后面的询问在前一个询问的基础上得到答案。

具体做法：

对于区间 $[i,i]$ 由于区间只有一个元素，我们很容易就能知道答案。然后一步一步从当前区间（已知答案）向下一个区间靠近。

我们设 $col[i]$ 表示当前颜色 i 出现了多少次 ans 表示当前共有多少种可行的配对方案（有多少种可以选到一双颜色相同的袜子），然后每次移动的时候更新答案——设当前颜色为 k 如果是增长区间就是 ans 加上 $C^2_{col[k]+1}-C^2_{col[k]}$ 如果是缩短就是 ans 减去 $C^2_{col[k]}-C^2_{col[k]-1}$

而这个询问的答案就是 $\frac{ans}{C^2_{r-l+1}}$

这里有个优化 $C^2_a = \frac{a(a-1)}{2}$

所以 $C^2_{a+1}-C^2_a = \frac{(a+1)a}{2} - \frac{a(a-1)}{2} = a$

所以 $C^2_{col[k]+1}-C^2_{col[k]} = col[k]$

算法总复杂度 $O(n\sqrt{n})$

下面的代码中 $deno$ 表示答案的分母(denominator) $nume$ 表示分子(numerator) sqn 表示块的大小 \sqrt{n} arr 是输入的数组， $node$ 是存储询问的结构体， tab 是询问序列（排序后的）， col 同上所述。

注意：由于 $++l$ 和 $--r$ 的存在，下面代码中的移动区间的 4 个 **for** 循环的位置很关键，不能改变它们之间的位置关系。

From: <https://wiki.cvbacm.com/> - CVBB ACM Team

Permanent link:

https://wiki.cvbacm.com/doku.php?id=2020-2021:teams:legal_string:%E5%90%84%E5%AD%A3%E5%BA%A6%E8%AE%AD%E7%BB%83%E8%AE%80%E5%BD%95%E6%99%AE%80%9A%E8%AB%E9%9F%E7%AE%97%E6%B3%95_lgrza&rev=1596468182

Last update: 2020/08/03 23:23