# 莫队算法

莫队算法是分块中一种经典模型，所谓莫队算法。就是对于很多次无修改的询问提出的。

设询问见存在递推关系，但是询问本身是无序的，且询问过程(除输入值外)其他参数不变。排序看具体情况，原则是尽可能把较大结果的中间态放前面

有以下解法:记录每组输入数据，对其按照一定顺序排序。使得正好能从第一个位置递推到最后一个位置，时间复杂度减小 (可能减到on也可能是介于暴力和on之间的复杂度)。

Hdu6333

<http://acm.hdu.edu.cn/showproblem.php?pid=6333>

题意是求sum(i=0;i<=m;i++ | C(n,i)) 设它是S(n,m) 1<=m<=n<=1e5

S(n,m)可以直接循环求，1e5的数据预处理阶乘逆元等。每次求C(n,i)用o1复杂度。所以求S(n,m)是m时间的  
但是吧，T等于1e5,总时间复杂度是T\*m=1e10 太慢；

而且经过我推算和谷歌上查资料，S(n,m)没有低于线性时间的求法，一般人的反应是求什么o(1)或者o(logn)的公式,很遗憾是不存在的，以后再遇到知道没有就好了。那这个方法行不通，要换一个思路了

考虑莫队离线，S(n,m)之间很容易知道存在递推关系：

S(n,m) =S(n,m-1)+ C(n,m) S(n,m) =2\*S(n-1,m)-C S(n-1,m)

公式二由组合数基本公式C(n,m)=C(n-1,m-1)+ C(n-1,m)得到，从杨辉三角容易看出

得到递推关系，也很容易想到，对输入的T组测试数据排序，使得较大的数据由较小的数据推来，相当于只算了较大数据，省去算较小数据的时间

怎么对数据排序，开始以为是先N后M的排序，然后从开始0递推就好

但这样虽然N从小到大，M确没有顺序，如果T组数据中M以：1 100000 1 100000 1 10000……这样一个大一个小的交替排列，每次都要过一遍1e5的递推，总时间会退化到1e10的。

解决这个问题就要对数据进行分块，比如k=500一块按照N分，什么意思。输入时用pos[N]代表N属于哪块，pos[N]=N/500代表对N进行每隔500分块

排序时按照先 分块编号 后M的顺序

一般每块区间k=sqrt(N),这样相率最佳，块数也是sqrt(N),每块最坏从0到M递推，时间是M\* sqrt(N) 。同时要有总和为T次的递推N，但每次只在块内进行，复杂度是T\* sqrt(N) 总时间是M\* sqrt(N)+ T\* sqrt(N) 约等于(1e5)^1.5

这道题当做典型的莫队算法例题

牛客网/洛谷1792

<https://ac.nowcoder.com/acm/contest/20322/J>

题意：,n个数，m此查询，查询静态区间不同数个数

//洛谷那个数据加强过就不能莫队了

牛客那个题，把区间翻倍，然后等价于查询[r,n+l]、

思路：分块，每块大小为sqrt(n)

把询问的左端点按照块编号排序,如果左端点在同一块里，就按照右端点排序(不是按照右端点所在块)，然后暴力寻找每个排序后的查询，最后输出时再排序回来，所谓暴力寻找，就是：建一个索引数组记录每个数组元素个数(如果数组元素很大可以离散化，但这两个题不需要)，然后维护一个总的cnt,这个cnt是当前区间不同数的个数，从第一个查询开始，从查的数据从左到右扫描即可，之后每个区间由第一个区间的结果转移而来，这个转移也是暴力进行的。

时间复杂度：O()

理论依据：对于原本询问，因为都是无序的随机数据，不论直接暴力还是从初始开始转移的方法，显然是n\*m复杂度。假如我们对询问进行排序，左端点按照端点编号排序,如果左端点编号相同，就按照右端点编号排序。此时如果大量数据是端点相同的，这种做法很快，极端点加入左端点全部相同，那么右端点按转移即可，复杂度O(n)。但这个办法处理交错数据时，比如询问是:[1,5],[2,6],[3,7],

[4,8]…

* 优化，奇数偶数优化，所有左端点在偶数号块的询问排序时，其右端点按照递增排序；所有左端点在奇数号块的询问排序时，其右端点按照递减排序。

理论依据：此时询问之间转移时，右端点正好尽可能相邻，可以少遍历一些数据。