

莫队

Mo's Algorithm

Idea: 对于一个长为 n 的序列，设有 m 次区间询问，则将原序列分割为大小为 $\frac{n}{\sqrt{m}}$ 的若干块。以左端点所在块为第一关键字、以右端点为第二关键字对询问排序，移动左右端点即可。

Complexity: $O(n\sqrt{m})$

Code:

```
1  int sq, belong[N];
2  struct Query{
3      int id, l, r, ans;
4      bool operator < (const Query &A) const{
5          return belong[l] == belong[A.l] ? r < A.r : belong[l] < belong[A.l];
6      }
7  }q[N];
8
9  inline void update(int pos, int val){
10     // update nowAns
11 }
12
13 int main(){
14     scanf("%d%d", &n, &m);
15     sq = n / sqrt(m);
16     for(int i = 1; i <= n; i++) belong[i] = (i - 1) / sq + 1;
17     for(int i = 1; i <= n; i++) scanf("%d", &a[i]);
18     for(int i = 1; i <= m; i++) scanf("%d%d", &q[i].l, &q[i].r), q[i].id = i;
19     sort(q+1, q+m+1);
20     for(int i = 1, l = 1, r = 0; i <= m; i++){
21         for(; l > q[i].l; l--) update(l-1, 1);
22         for(; r < q[i].r; r++) update(r+1, 1);
23         for(; l < q[i].l; l++) update(l, -1);
24         for(; r > q[i].r; r--) update(r, -1);
25         q[i].ans = nowAns;
26     }
27     sort(q+1, q+m+1, [&](const Query &A, const Query &B){ return A.id < B.id; });
28     // output
29     return 0;
30 }
```