

# 分块

Idea: 将长度为  $n$  的序列分为长度为  $\sqrt{n}$  的  $\sqrt{n}$  块, 针对每一块维护一些信息。修改或询问时, 暴力修改/询问块外的数, 利用维护的信息修改/询问块内的数。

Complexity:  $O(n\sqrt{n} + q\sqrt{n})$

Code:  $L[i], R[i]$  分别表示块  $i$  的左右端点位置;

$belong[i]$  表示第  $i$  个数所属块的编号;

$sq$  为一个块的大小, 一共有  $belong[n]$  个块。

```
1  int belong[N], sq;
2  int L[SQRN], R[SQRN];
3  inline void init(){
4      sq = sqrt(n);
5      for(int i = 1; i <= n; i++)
6          belong[i] = (i - 1) / sq + 1;
7      for(int i = 1; i <= belong[n]; i++){
8          L[i] = (i - 1) * sq + 1;
9          R[i] = i * sq;
10     } R[belong[n]] = n;
11 }
12 inline void maintain(int x){
13     // maintain info of block x after bf
14 }
15 inline void modify(int l, int r, int val){
16     for(int i = l; i <= min(R[belong[l]], r); i++){
17         // modify left-side part with bf
18     }
19     maintain(belong[l]);
20     if(belong[l] != belong[r]){
21         for(int i = L[belong[r]]; i <= r; i++){
22             // modify right-side part with bf
23         }
24         maintain(belong[r]);
25     }
26     for(int i = belong[l] + 1; i < belong[r]; i++){
27         // modify the whole block
28     }
29 }
30 inline int query(int l, int r, int x){
31     int res = 0;
32     for(int i = l; i <= min(R[belong[l]], r); i++){
33         // query left-side part with bf
34     }
35     if(belong[l] != belong[r]){
36         for(int i = L[belong[r]]; i <= r; i++){
37             // query right-side part with bf
38         }
39     }
40     for(int i = belong[l] + 1; i < belong[r]; i++){
41         // query the whole block
42     }
43     return res;
44 }
45
46 int main(){
47     // input
48     for(int i = 1; i <= belong[n]; i++) maintain(i);
49     // ...
50 }
```