

```
1
2 class Bucket {
3 public:
4     int size;
5     vector<long long> v;
6     long long addSum;
7     long long stageNum;
8
9     Bucket(int n) {
10         size = n;
11         v.resize(n);
12         addSum = 0;
13         stageNum = 0;
14     }
15
16     // add a [l, r)
17     void add(int l, int r, long long a) {
18         if (l <= 0 && r >= size) {
19             addSum += a;
20         } else {
21             for (int i = max(0, l); i < min(size, r); i++)
22                 v[i] += a;
23         }
24     }
25
26     // add [l, r)
27     int addStage(int l, int r, long long a) {
28         if (l <= 0 && r >= size) {
29             stageNum++;
30             add(l, r, a);
31             return size;
32         } else {
33             int c = 0;
34             for (int i = max(0, l); i < min(size, r); i++) {
35                 v[i] += a + c++;
36             }
37             return c;
38         }
39     }
40
41     long long get(int i) {
42         if (stageNum > 0) {
43             for (int i = 0; i < size; i++)
44                 v[i] += stageNum * i;
45             stageNum = 0;
46         }
47         if (addSum > 0) {
48             for (int i = 0; i < size; i++)
49                 v[i] += addSum;
50             addSum = 0;
```

```
51     }
52     return v[i];
53 }
54 };
55
56 const int BSIZE = 300;
57
58 class Buckets {
59 public:
60     vector <Bucket> b;
61     int size;
62     int bsize;
63     int num;
64
65     Buckets(int n) {
66         size = n;
67         bsize = (n + BSIZE - 1) / BSIZE;
68         for (; n > 0; n -= bsize) {
69             Bucket c(min(n, bsize));
70             b.push_back(c);
71         }
72         num = b.size();
73     }
74
75     void add(int l, int r, long long a) {
76         for (int i = 0; i < num; i++)
77             b[i].add(l - bsize * i, r - bsize * i, a);
78     }
79
80     int addStage(int l, int r) {
81         int c = 0;
82         for (int i = 0; i < num; i++) {
83             c += b[i].addStage(l - bsize * i, r - bsize * i, c);
84         }
85         return c;
86     }
87
88     long long get(int i) { return b[i / bsize].get(i - (i / bsize) * bsize); }
89 };
90
```