

```
i = (c1 == cn) {
    for (int i = 1; i < n; i++) {
        sortUpd(p[i], x);
    }
}
```

Н-Бем

```
{ for (int i = 1, j = (c1 + 1) * K; i < j, i++) {
```

```
{ for (int i = c1 + 1; i < cn; i++) {
    b[i] += x
}
```

{

```
for (int i = cn * K; i < r; i++) {
    sortUpd(p[i], x);
}
```

{

```
{ long long (int l, int r) {
    long res = sortGet(l, r);
    for (int i = l; i < r; i++) {
        res += b[i] * sortGet(i, r);
    }
}
```

return res;

```
{ void push (int t) {
    int l = b[t];
}
```

```
{ int t = t * K, m = min(n, (t + 1) * K);
    for (int i = l; i < r; i++) {
        sortUpd(p[i], b[i]);
    }
}
```

b[t] = 0;

```
{ void exchange (int a, int b) {
```

int ca = a / K, cb = b / K

if (ca != cb) {

push (ca);
push (cb);

int (p[a] < p[b]) {

sortUpd (ca, p[a], p[b] - 1);
sortUpd (cb, p[b] - 1, 1);