

3-бөлім

```

if (c1 == cn) {
    for (int i = 1; i <= n; i++) {
        P[id][i] += x;
    }
    return;
}
for (int i = 1, j = (c1 + 1) * K; i < j; i++) {
    P[id][i] += x;
}
for (int i = c1 + 1; i <= cn; i++) {
    Q[id][i] += x;
}
for (int i = cn * K; i <= n; i++) {
    P[id][i] += x;
}
    
```

```

int sqrt Get (int l, int r) {
    int cl = l / K, cn = r / K;
    int ret = 0;
    if (cl == cn) {
        for (int i = l; i <= r; i++) {
            ret += S[i];
        }
        return ret;
    }
    for (int i = cl, j = (cl + 1) * K; i < j; i++) {
        ret += S[i];
    }
    for (int i = cl + 1; i <= cn; i++) {
        ret += T[i];
    }
    for (int i = cn * K; i <= r; i++) {
        ret += S[i];
    }
    return ret;
}
    
```

```

int sqrt Get (int id, int p) {
    return P[id][p] + Q[id][p / K];
}
int sqrt Get (int id, int l, int r) {
    return sqrt Get (id, r) - (l ? sqrt Get (id, l - 1) : 0);
}
    
```

```

void build () {
    for (int i = 0; i * K <= n; i++) {
        int l = i * K, r = min(n, (i + 1) * K);
        for (int j = l; j <= r; j++) {
            int x = p[j];
            int id = x / K;
            P[id][x] += P[id][min(n, (id + 1) * K)];
            Q[id][id + 1] += a[id][id + 1];
        }
        for (int j = l; j <= r; j++) {
            P[j][j] += P[j][j - 1];
        }
        for (int j = l; j <= r; j++) {
            Q[j][j] += Q[j][j - 1];
        }
    }
}
void update (int l, int r, int v) {
    int cl = l / K, cn = r / K;
    }
    
```

```

int sqrt Get (int id, int p) {
    return P[id][p] + Q[id][p / K];
}
int sqrt Get (int id, int l, int r) {
    return sqrt Get (id, r) - (l ? sqrt Get (id, l - 1) : 0);
}
    
```

```

int sqrt Get (int id, int p) {
    return P[id][p] + Q[id][p / K];
}
int sqrt Get (int id, int l, int r) {
    return sqrt Get (id, r) - (l ? sqrt Get (id, l - 1) : 0);
}
void build () {
    for (int i = 0; i * K <= n; i++) {
        int l = i * K, r = min(n, (i + 1) * K);
        for (int j = l; j <= r; j++) {
            int x = p[j];
            int id = x / K;
            P[id][x] += P[id][min(n, (id + 1) * K)];
            Q[id][id + 1] += a[id][id + 1];
        }
        for (int j = l; j <= r; j++) {
            P[j][j] += P[j][j - 1];
        }
        for (int j = l; j <= r; j++) {
            Q[j][j] += Q[j][j - 1];
        }
    }
}
void update (int l, int r, int v) {
    int cl = l / K, cn = r / K;
}
    
```