Map写法：

#include <iostream>

#include <cstring>

#include <algorithm>

#include <map>

using namespace std;

const int INF = 2e9;

int n, x, y, z;

int main()

{

map<int, int> b;

scanf("%d%d%d%d", &n, &x, &y, &z);

for (int i = 0; i < n; i ++ )

{

int l, r;

scanf("%d%d", &l, &r);

b[-INF] += x;

b[l] += y - x;

b[r + 1] += z - y;

b[INF] -= z;

}

int res = 0, sum = 0;

for (auto& [k, v]: b)

{

sum += v;

res = max(res, sum);

}

printf("%d\n", res);

return 0;

}

手动写法：

#include <iostream>

#include <cstring>

#include <algorithm>

#include <vector>

using namespace std;

const int N = 20010, INF = 2e9;

int n, x, y, z;

vector<int> xs;

int l[N], r[N], b[N \* 2];

int find(int v)

{

int l = 0, r = xs.size() - 1;

while (l < r)

{

int mid = l + r >> 1;

if (xs[mid] >= v) r = mid;

else l = mid + 1;

}

return r;

}

int main()

{

scanf("%d%d%d%d", &n, &x, &y, &z);

xs.push\_back(-INF), xs.push\_back(INF);

for (int i = 0; i < n; i ++ )

{

scanf("%d%d", &l[i], &r[i]);

xs.push\_back(l[i]);

xs.push\_back(r[i] + 1);

}

sort(xs.begin(), xs.end());

xs.erase(unique(xs.begin(), xs.end()), xs.end());

for (int i = 0; i < n; i ++ )

{

int L = find(l[i]), R = find(r[i] + 1);

b[0] += x;

b[L] += y - x;

b[R] += z - y;

b[xs.size() - 1] -= z;

}

int res = 0, sum = 0;

for (int i = 0; i < xs.size(); i ++ )

{

sum += b[i];

res = max(res, sum);

}

printf("%d\n", res);

return 0;

}