**Solution**

We only need to check all left\_points and right\_point+1.

Left is point of coming, right+1 is point of leaving.

Let d(i) be the max with last point at chk[i].

d[i]=max{d[j]+w(set(i)-set(j)): for all j<i}, where set(i) is the set shot by chk[i].

For i<k, let p[i]=d[i]-w(i,k), where w(i,k)=weight of segments containing [i,k].

When sweeping to k, we keep p[i], and then   
d[k]=max{p[i]: for all i<k}+w(set[k]).

When k=seg.right+1, seg is out of date, do add w(seg) for interval [seg.left,seg.right];

when k=seg.left, seg is coming, do add –w(seg) for its interval.

Using a segment tree, and it can be done in O(nlogn)

An O(n^2) slower algorithm is used to check the solution.

Two greedy algorithms are used to defense wrong algorithms.