**區間的線段**

Solution：

This task combines the following two classic problems:

1. Finding the max weight subarray in an array. It can be solved by a linear time DP.
2. Finding the maximum clique of an interval graph (i.e. , finding a point intersecting the largest weight of intervals). It can be solved by a sweep-line algorithm.

The first observation is that we need only consider all left-points and all (right-point+1). Reason: if neither the two cases, we can move to the left without changing the result.

Let F(j) be the maximized total weight with interval [BestLeft(j), j]. Since BestLeft(j) is either j or <j, we have

F(j)=max{ p(j,j), p(BestLeft(j-1),j)}, where p(i,j) is the weight intersected by [i , j].

Not hard to see that

p(BestLeft(j-1),j)=F(j-1)+(weight of intervals starting at j).

Algorithm:

Preprocessing: put all L[i] into add[], R[i]+1 into rem[], and both into check[];

Sort the three arrays

Sweep on check points. For each check[j],

compute p(j,j) from p(j-1,j-1);

compute F(j) from F(j-1)

Time=O(nlogn)

Not so-efficient algorithms：

Subtask1 : check all possible ranges when M is small. O(NM^2)

Subtask2 : check all pairs of check points O(N^3)

Subtask3 : check all pairs of check points O(N^2)

Subtask4 : for the degenerated case that each segment is a point O(NlogN)

Subtask5 : O(NlogN)

Files

Rangehit\_xx.in test data for subtask xx

Rangehit\_xx.out output subtask xx

Rangehit.c program solving all subtasks

Rangehit\_R2.c program with time complexity nR^2

Rangehit\_n3.c program with time complexity n^3

Rangehit\_n2.c program with time complexity n^2

Rangehit\_n2b.c another program with time complexity n^2

Rangehit\_s4.c program for subtask 4, time complexity nlogn but all segments are points