前処理

アルゴリズム論

中野

Note 9　幾何情報の利用

2020.5.14 作成

20206.20 update 6.22　2021.5.31

Range Searching Count

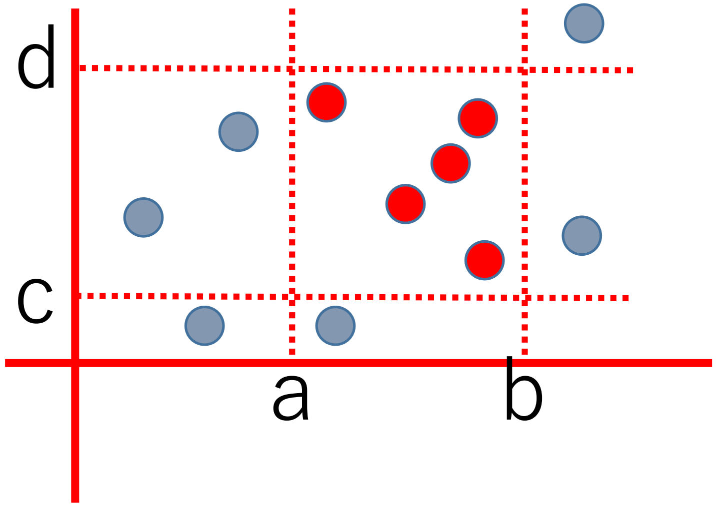
(準備)平面上の点の座標を(x,y)のように表す。xはx座標、yはy座標である。

Range Searching Count
入力　平面上のn点と軸平行な長方形
出力　何点が長方形の内部にあるかな？

長方形の外周は x=a, x=b, y=c, y=d上の線分であるとする.
点(x,y)は a≦x≦b かつ c≦y≦dのときのみ長方形の内部にあるとする.

(データベースのmultikey searchingの簡単な例)

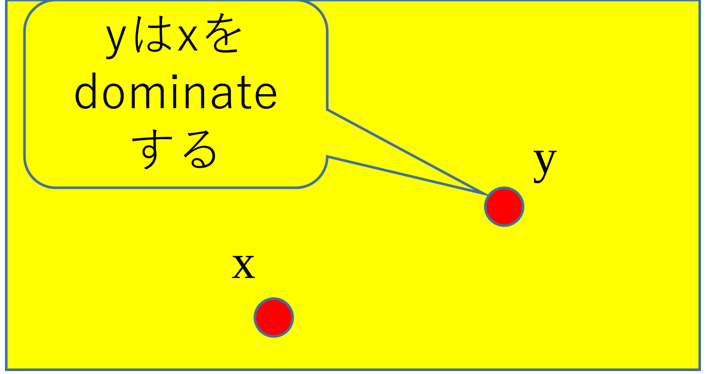
簡単な方法
各点について
a≦x≦b かつ c≦y≦dであるかどうか判定すればいいので O(N)時間で解が求まる.
これ以上に改善できるかな？



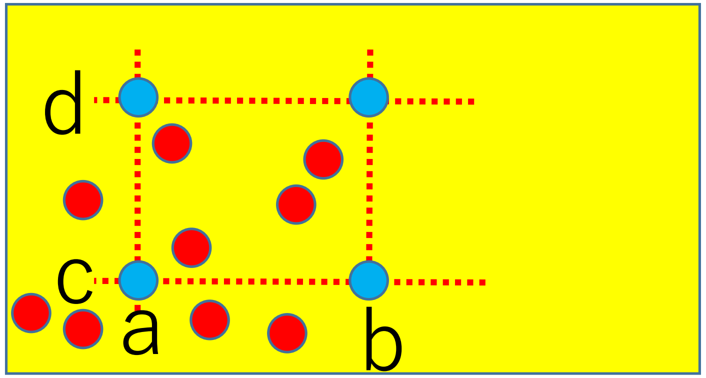
count = 0
for 各点p_i
　if　a≦x(p_i) ≦b　かつ　c≦y(p_i)≦d
　　count = count +1

前処理を使えばどうかな？
平面上のn点は固定しておく. いろいろな長方形が与えられたとき,
長方形の内部になる点の個数を求めたい.

Vector dominance法



点(x1, y1)と点(x2, y2)があるとする。もし, x1≦x2かつy1≦y2ならば
点(x2, y2)は点(x1, y1)をdominate(支配)するという。（南西にある点をdominateする。）
点pがdominateする点の個数をQ(p)で表す。



長方形の南西端の点は(a,c), 南東端の点は(b,c), 北西端の点は(a,d),
北東端の点は(b,d)であるとする。
この長方形の内部にある点の個数はQ((b,d))-Q((a,d))-Q((b,c))+Q((a,c))である。

よってRange Searching Count問題は, Dominateする点の個数を求める問題に帰着できた。

n個の各点について, その点を通る水平線と垂直線を描こう. 平面は$(n+1)^2$個の小さな長方形に分割される。
任意の小さな長方形をひとつ選ぶ。小さな長方形の内部の任意の2点p, qについて Q(p)=Q(q)である。
つまり平面上には無限個の点があるが, Qに関する有限個の同値類に分割できた！

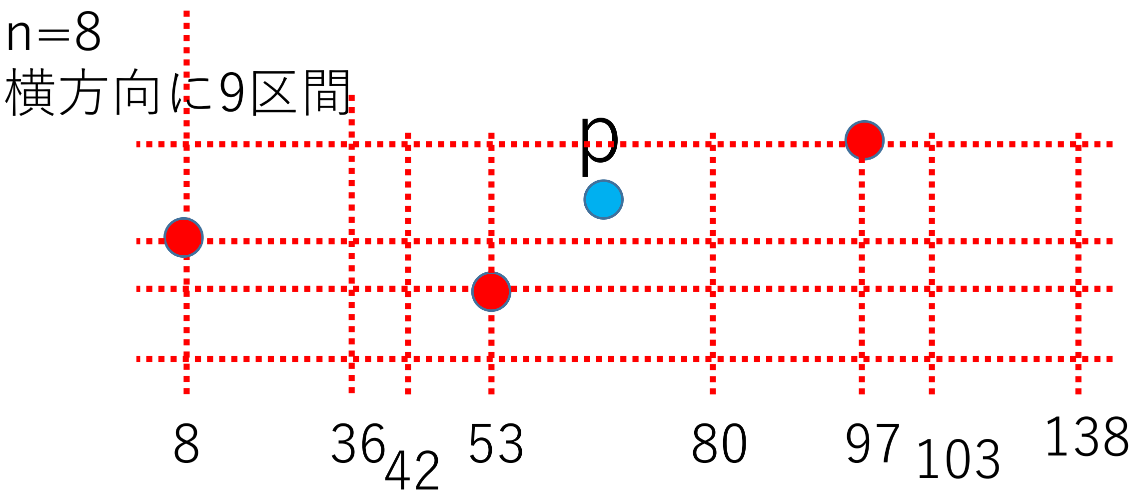
$(n+1)^2$個の長方形のそれぞれについて、内部の点pのQ(p)の値を前処理として計算しておこう。
($O(n^3)$時間, もしくは DPを使って$O(n^2)$時間でできる。)

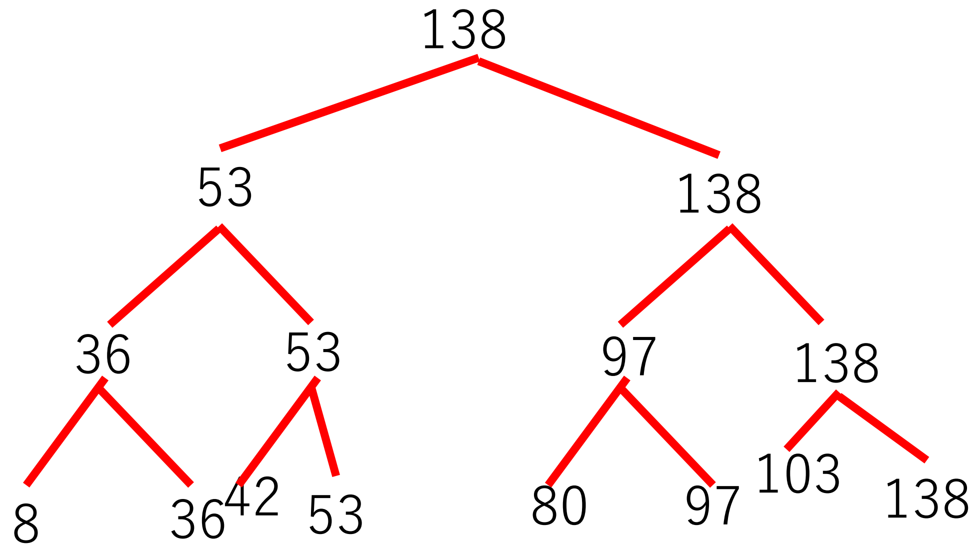
Vector dominance法

点pが与えられたとき, 2分探索により
$(n+1)^2$個の長方形の内の
どの長方形にpが含まれるかだけを調べれば,
Q(p)の値がわかる.
(O(log n)時間でできる.)
(この2分探索用の木を縦方向、横方向の２本前処理で作成する。
さらに O(n log n)時間必要.)(ソートにO(n log n) 木の作成にn)

以上により
前処理に $O(n^2)$ + O(n log n)時間
すなわち$O(n^2)$時間かかる.
必要なメモリは $O(n^2)$である.

ひとつの長方形が与えられたとき 四隅の4点について
縦方向の区間と横方向の区間を求め、その区間の
Qの値を求め, Q((b,d))-Q((a,d))-Q((b,c))+Q((a,c))を求める.
これは, O(log n)時間で計算できる.



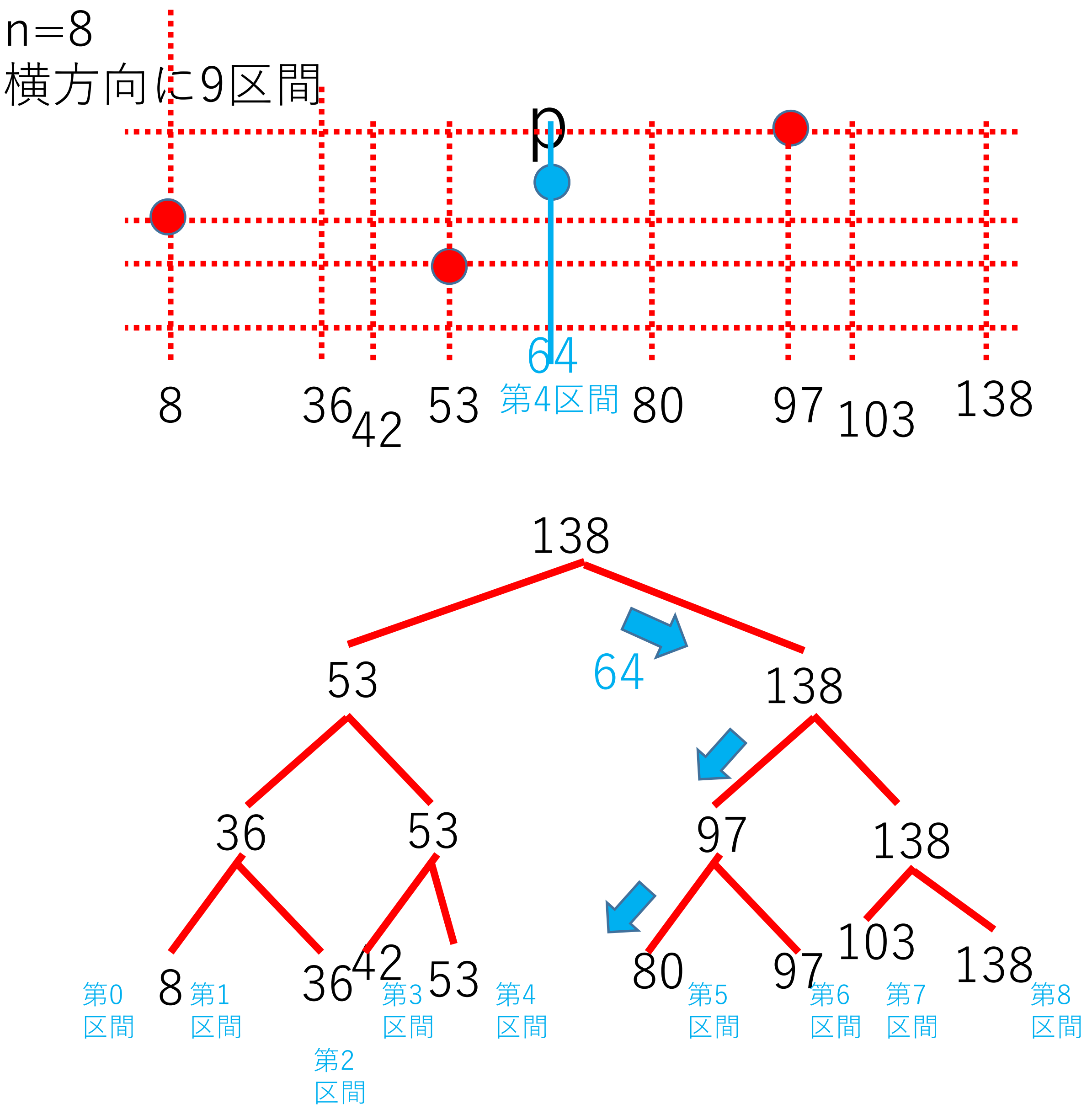


Vector dominance法

点pが与えられたとき, 2分探索により
$(n+1)^2$個の長方形の内の
どの長方形にpが含まれるかだけを調べれば,
Q(p)の値がわかる.
(O(log n)時間でできる.)
(この2分探索用の木を縦方向、横方向の２本前処理で作成する。
さらに O(n log n)時間必要.)(ソートにO(n log n) 木の作成にn)

以上により
前処理に $O(n^2)$ + O(n log n)時間
すなわち$O(n^2)$時間かかる.
必要なメモリは $O(n^2)$である.

ひとつの長方形が与えられたとき 四隅の4点について
縦方向の区間と横方向の区間を求め、その区間の
Qの値を求め, Q((b,d))-Q((a,d))-Q((b,c))+Q((a,c))を求める.
これは, O(log n)時間で計算できる.



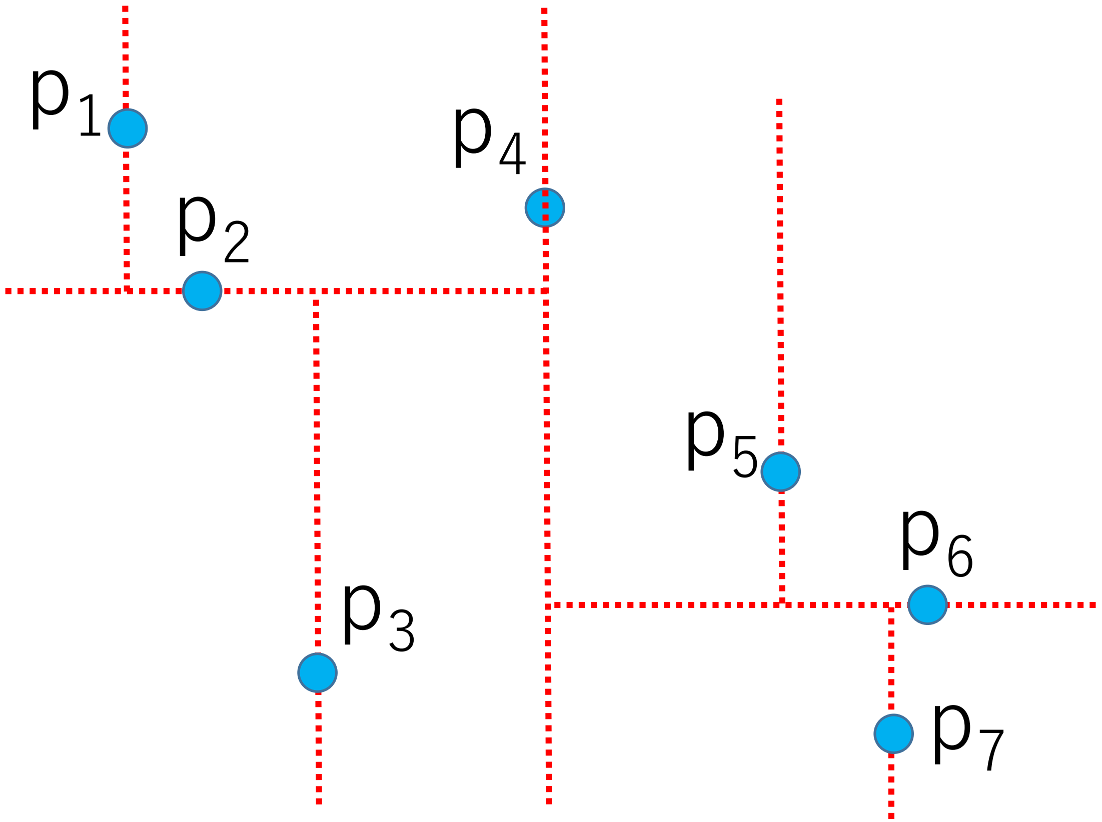
もっと改善できないかな?

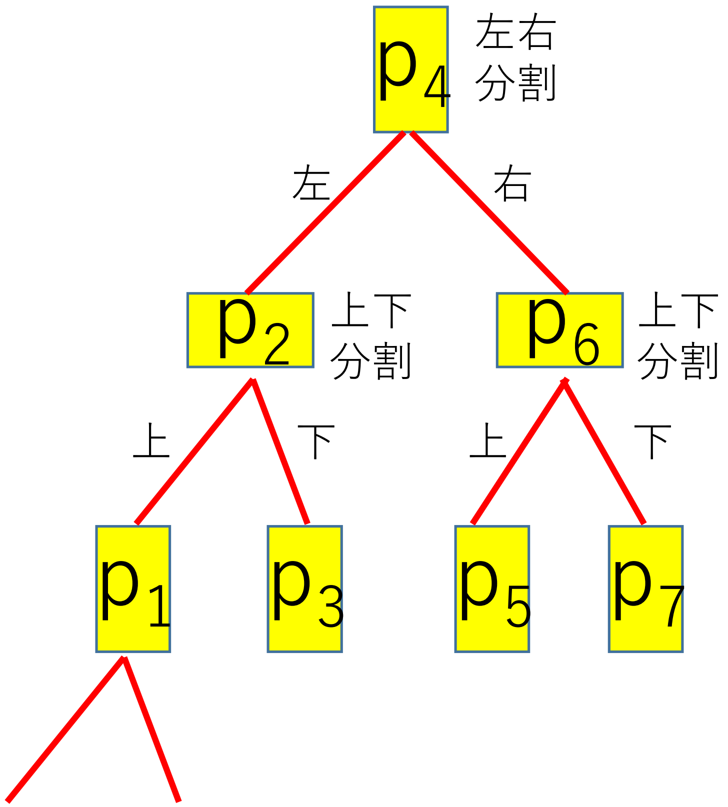
平面上のn点がdynamicに変化する場合(insert, deleteがある場合)はどうかな?

countingでなくreporting問題

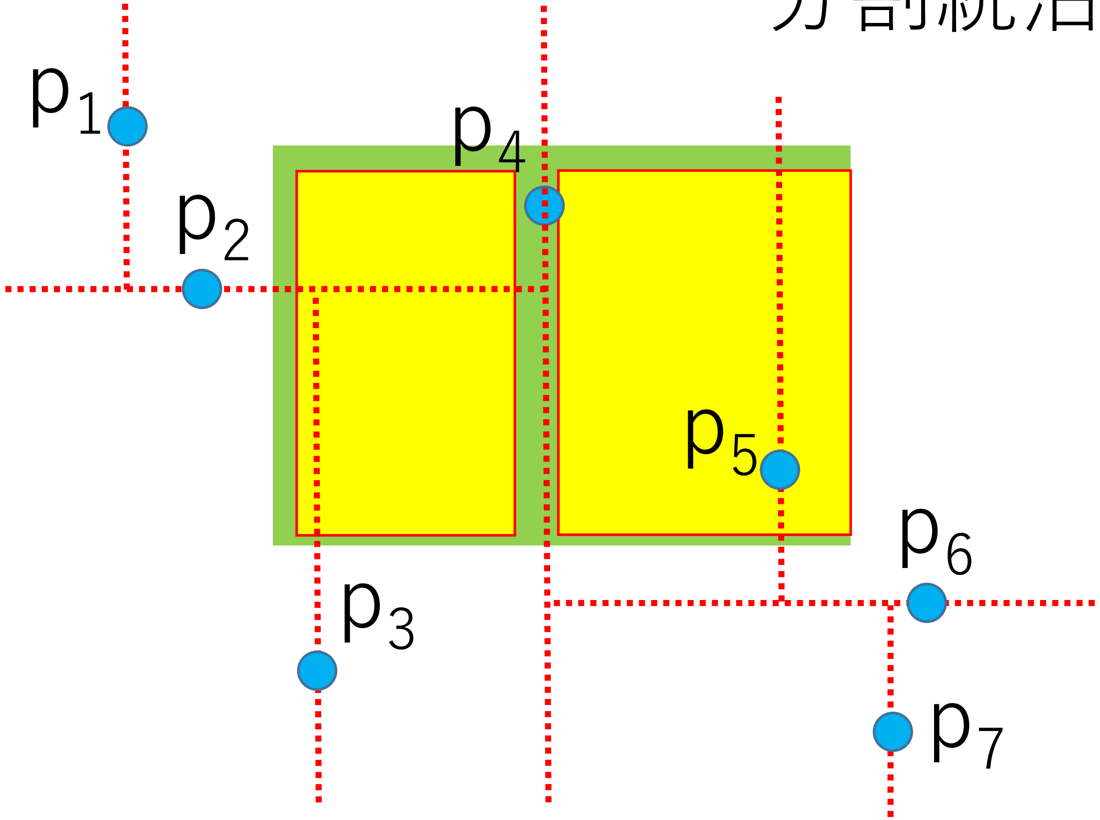
(長方形内の点をすべて出力する)を解くには？

Range Searching Report

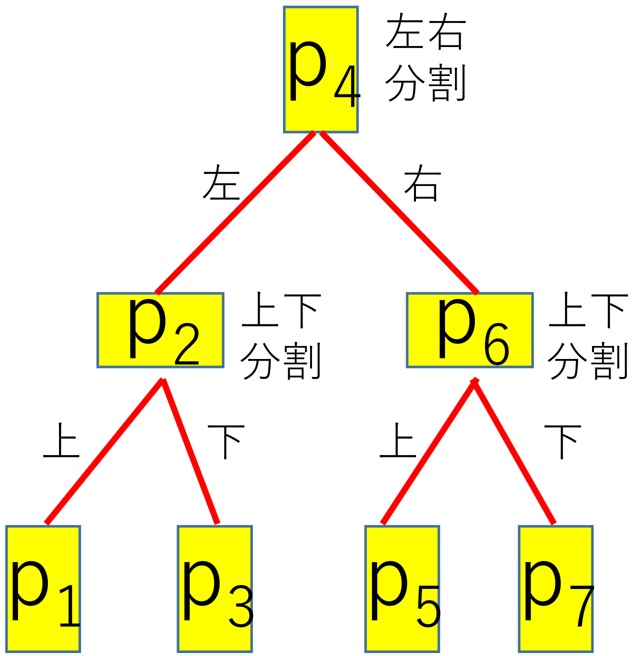




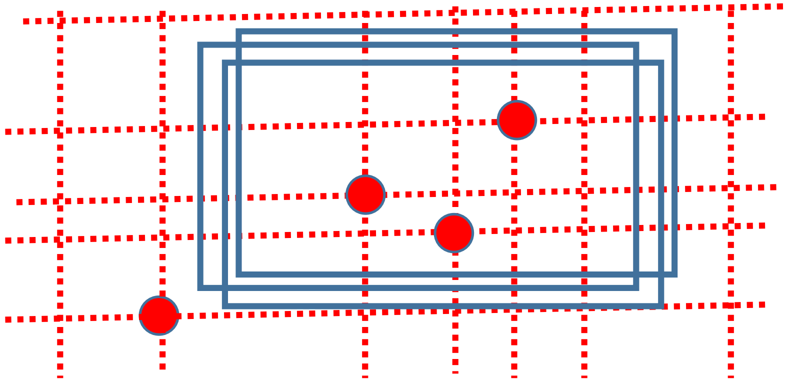
Range Searching Report （つづき）



分割統治法



メモリをたくさん使う方法



長方形をRとする.
Rの南西端の点が(a,c), 北東端の点が(b,d)であるとする.
先の問題のように平面を$(n+1)^2$個の長方形に分割する.
長方形の南西端の点を(a,c)と同じ長方形内で移動する限り,
Range Searching Reportの解は同じである.
同様に, 長方形の北東端の点を(b,d)と同じ長方形内で移動する限り,
Range Searching Reportの解は同じである.
つまり, 解は$(n+1)^4$個の同値類に分割できる.
解は最大n点からなる。

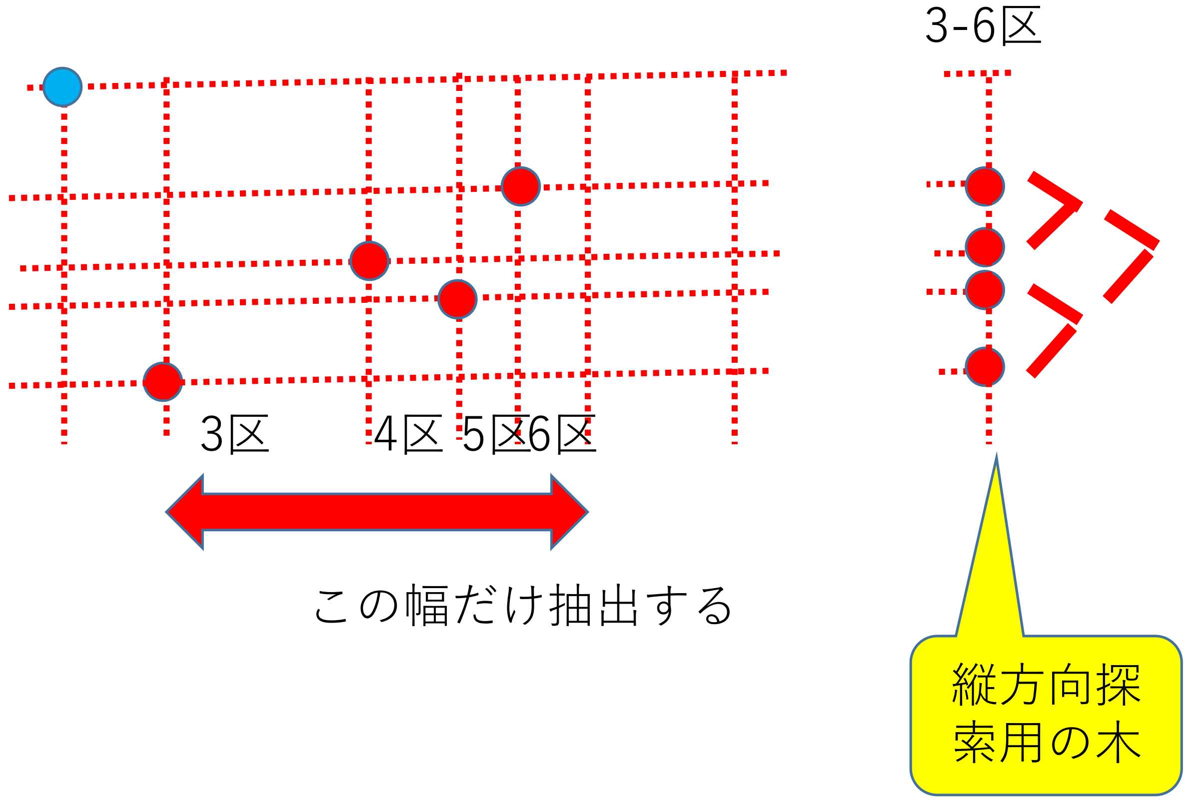
必要なメモリは$O(n^5)$である.
前処理にかかる時間は？　$O(n^5)$
長方形が与えられたときに, 解を求めるのに必要な時間は？　O(log n)時間
(縦方向は第？区間から第？区間まで　　横方向は第？区間から第？区間までを計算。 O(4 log n)時間！
あとは解の表(エントリn^4個)をひくだけ。解の出力にO(n)時間かかるけど

メモリを節約する方法

前ページの方法をx方向についてのみ行なう.
つまり, 与えられる長方形のx座標に応じて
高々$(n+1)^2$種類の問題のみがある.

これらの各問題について
1次元のRange Searching Reportを行う.
各問題について専用の(縦方向の)
2分探索用の配列を用意しておく.

必要なメモリは$O(n^3)$である.
前処理にかかる時間は？　$O(n^3)$

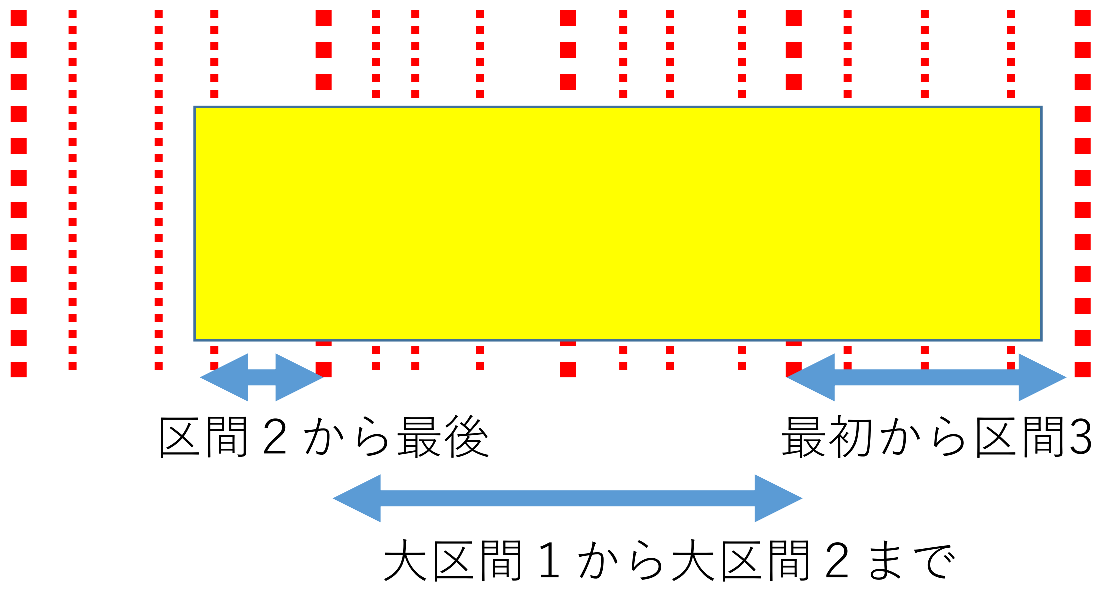


長方形が与えられたときに, 解を求めるのに必要な時間は？　$O(\log n)$時間
(横方向は第？区間から第？区間まで　　あとは縦方向の１次元の問題を解くだけ。

(もっと) メモリを節約する方法

x方向の問題の分割を
2段階でおこなう。。。
√n ごとに大まかに区切る
のち細分する。

(略)



メモリ $O(n^2)$ を２個
長方形が与えられたときに,
解を求めるのに必要な時間は $O(\log n)$時間