#1

For 2 database A and B, compare the medians of A and B. (k = ½ n). Suppose B[k]>A[k], otherwise we can have same conclusion. The elements greater than B[k] and the elements lower than A[k] can be ignore. We can get new A and B set without those invalid elements.

Again, we do the same things until there are only 2 elements.

Median(n,a,b)

If n=1 then return min(A(a+k),B(b+k))\

K=

If A(a+k)<B(b+k)

Then return median(k,a+,b)

Else

return median(k,a,b+)

#3

let S be the card set, devide set into 2 group, first of and rest. Run algorithm on two sedes sperately.

F(S,n)

{if |S|=1 then return the card

if |S|=2 then test 2 card and return one if equal

S1 = S[first ]

S2 = S[rest]

Recard1 = F(S1)

if recard1 has value

then test the value with all other cards

if no card have majority equivalence

then recard2 = F(S2)

then test recard2 with all other cards

return card with majority equivalence

}

#5

label the lines in order of increasing slope. Let m=. We recursively compute the sequence of visible lines among L1 --- Lm. We also compute the sequence of points a1 --- ap-1 where ak is the intersection of 2 lines. Like this, we computer sequence of visible lines among Lm+1 --- Ln, together with sequence of intersection points bk. When combining two results, we merge two lists into a new single list.

#add1

Iterate the numbers for the first bits, and divide the array to two parts by the bits(0 or 1)

repeat the process to rest bits. Each time search one bit and divide the array.

The complexity of the algorithm is N + N/2 +… +1 = 2N

#add2



