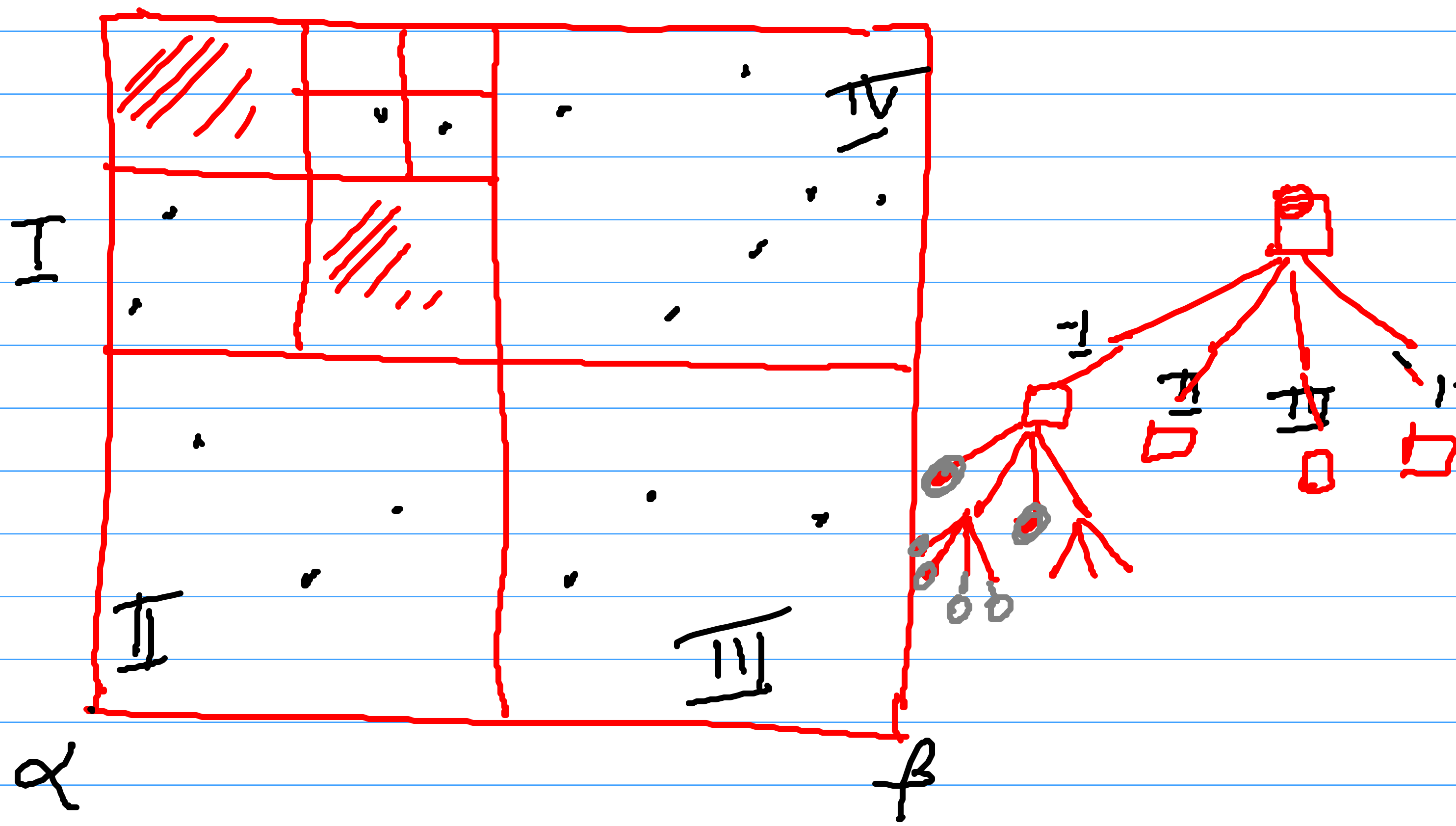
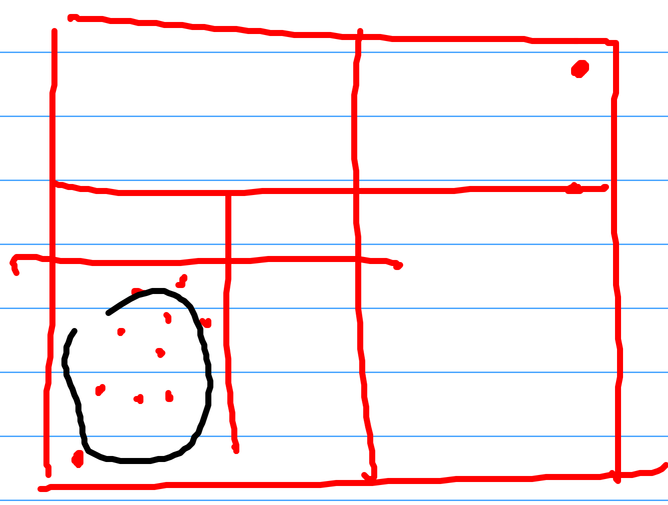
Computational Geometry CSL 852
Lecture 30

Topic : Quadtrees $\rightarrow \epsilon$-WSPD



Quadtree : ht ?

$\log |\text{spread}|$



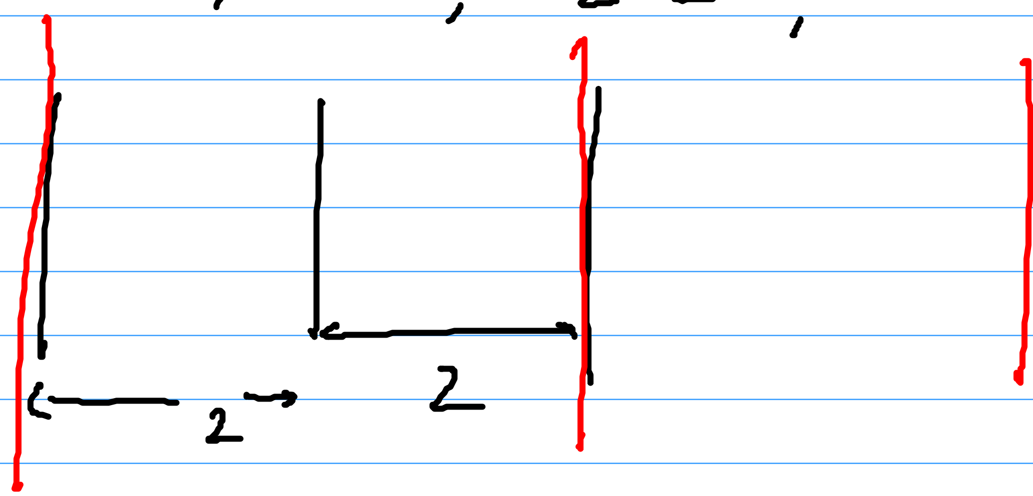
Canonical Quadtree

If gridpoints have some specific values, say powers of 2, then it is a canonical grid. We want to align our quadtree with such a family grids

G_i : $0, \; 2^i, \; 2\cdot 2^i, \; \ldots$

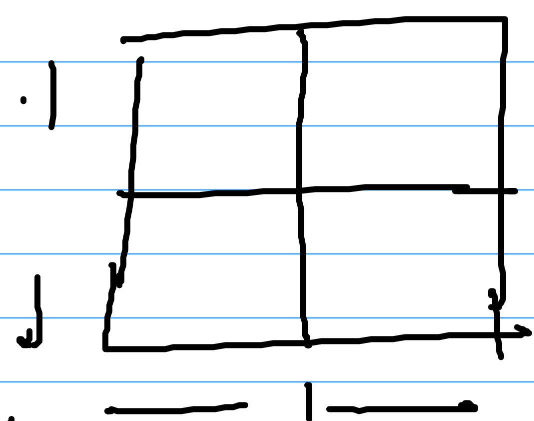
G_1

G_2

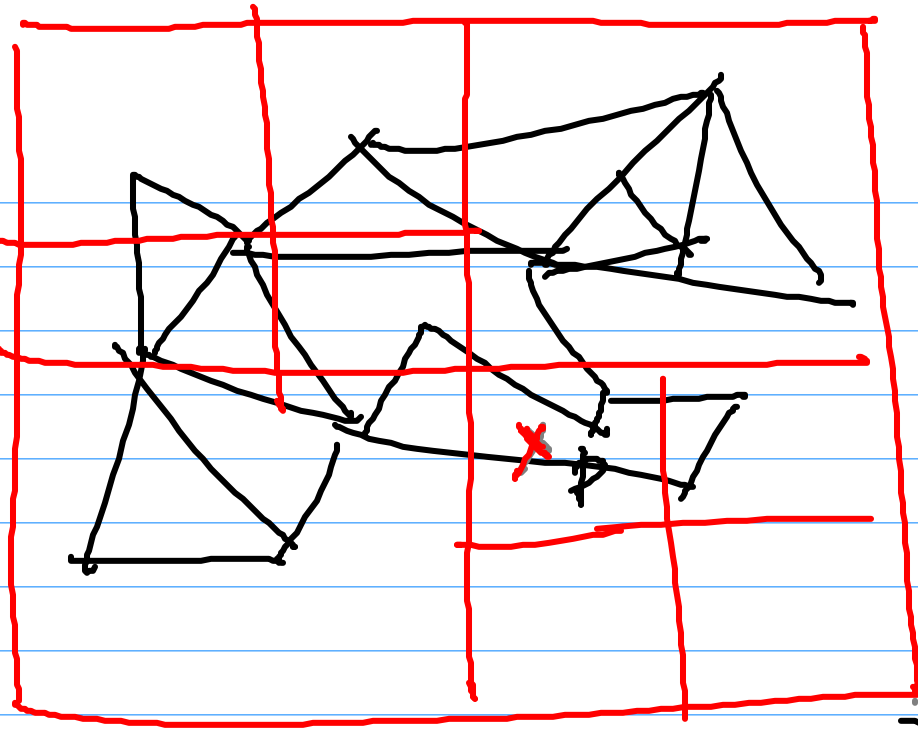


$G_i \subset G_j \quad j < i$

G_i : $\frac{1}{2^i}$



<u>Advantage</u>: At depth i from root, we know exactly which subsquares can be present. $\Rightarrow$ Can be hashed.



point location

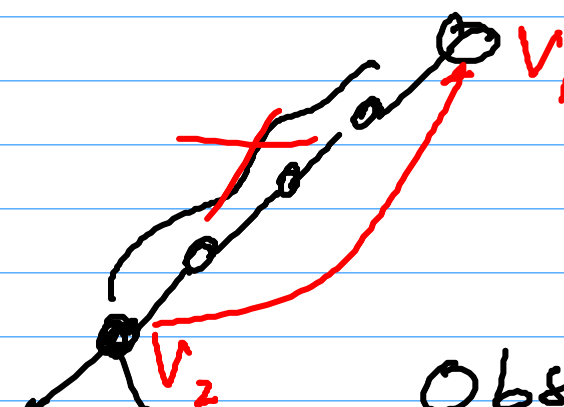
which is the smallest subsquare i.e. largest j, for which G_j contains p

Query : the ht of quadtree

Space :

} Unbounded

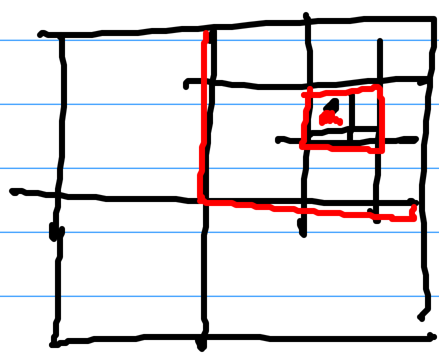
Observation : Compress the nodes having only one child



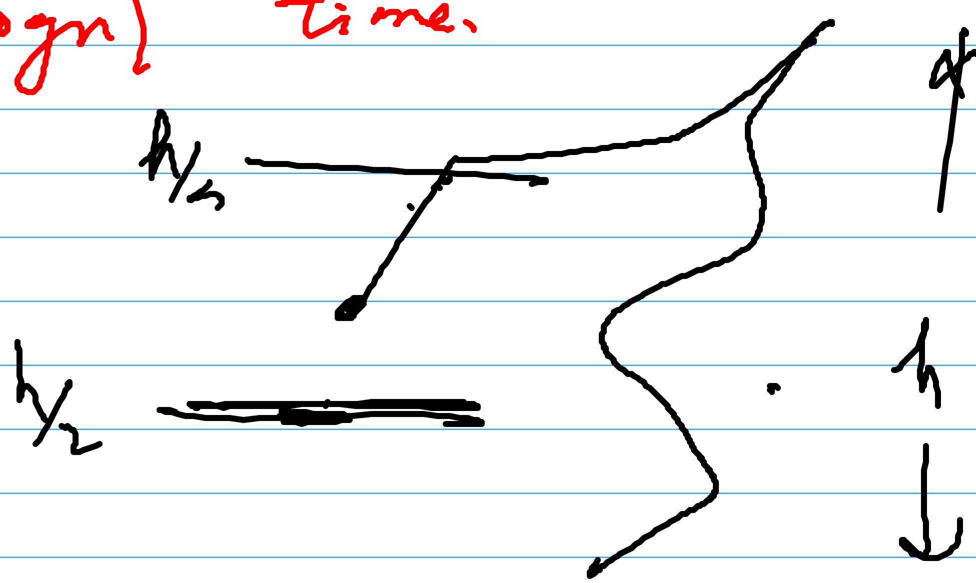
Observation : A compressed quadtree of n points has atmost $\boxed{O(n)}$ nodes.

By hashing: we mean a performance $O(n)$ space
$O(1)$ search time
(eg. Universal hash functions)

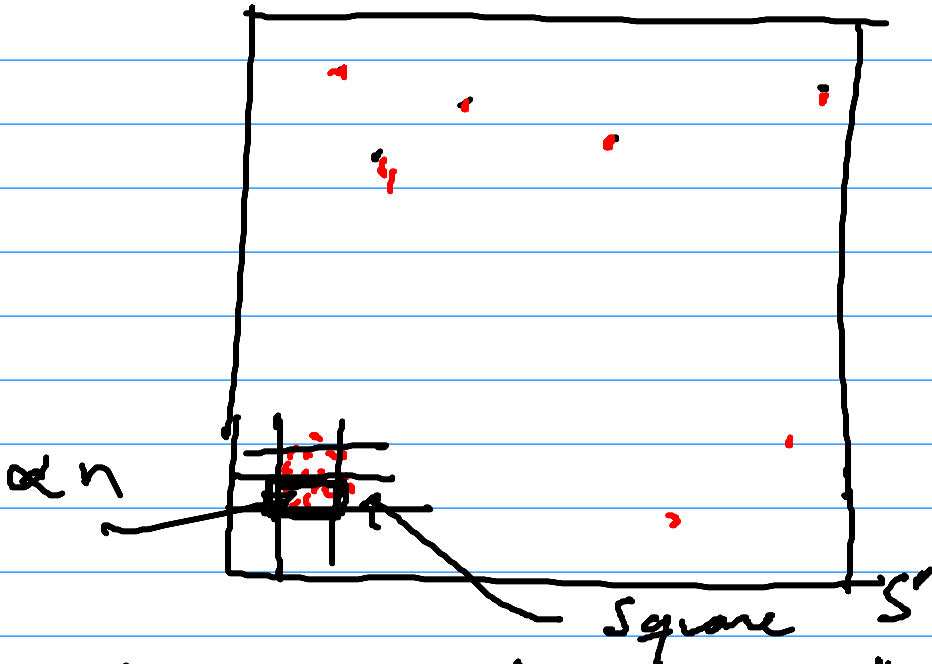
Speed up point location using "binary search" on the search path.

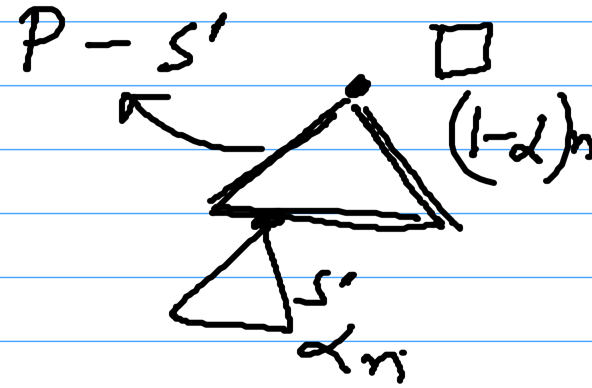


Claim: In a compressed quadtree, we can do point location in $O(\log n)$ time.



Constructing a Quadtree efficiently

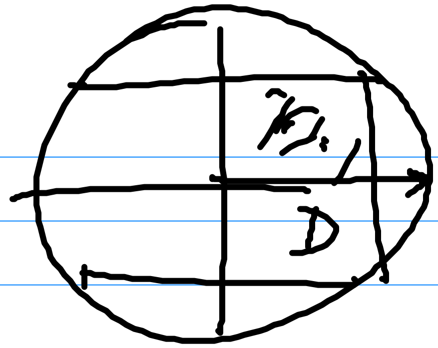




Can we find a "canonical subsquare" that contains a large fraction of the n points, i.e. αn for some constant $0 < \alpha < 1$?

$$T(n) = T(\alpha n) + T((1-\alpha)n) + \text{time to find the dense subsquare } O(n)$$

<u>Claim</u>: Given a point set P of n points, we can compute a disk D that contains K points $(K \leq n)$ such that radius $D \leq 2\, R_{opt}(K)$



D can be found in $O\left(n \cdot \left(\frac{n}{k}\right)^2\right)$ steps

If $k = \alpha n \sim O(n)$

<u>Claim</u> : Compressed Quadtree can be constructed in $O(n \log n)$ time.