CS428: Information Processing for Sensor Networks

Information Management II:

Geometric Querying,
Kinetic Data Structures,
Mobile Clustering

May 15, 2003

- **Geometric Range Searching:** Jiri Matousek, *Computing Surveys,* vol 26, number 4, page. 422-461, 1994.

- **Motion:** Leonidas Guibas, *Computational Geometry Handbook,* CRC Press, to appear, 2003.

- Presented by: Anne Collins

Geometric Range Searching

- Given a collection **P** of n points in E^d, and a family **R** of ranges $R \subseteq \mathrm{E}^d$, preprocess **P** to enable efficient queries

- Typical queries include:
  - Counting elements in **P** $\cap R$,
  - Finding max/min element of **P** $\cap R$,
  - Reporting all elements in **P** $\cap R$,

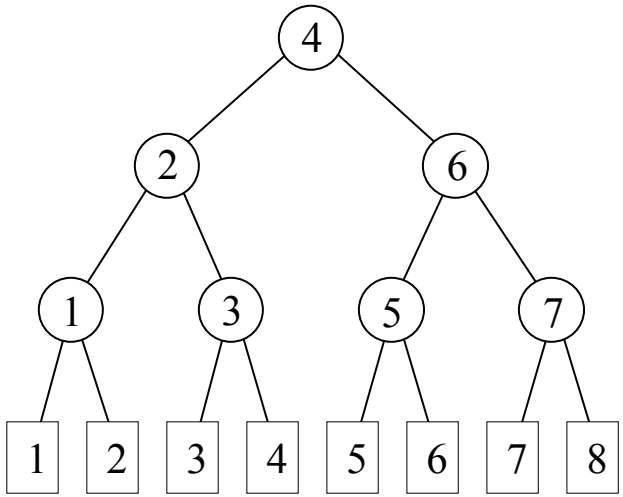
Query Performance Measures

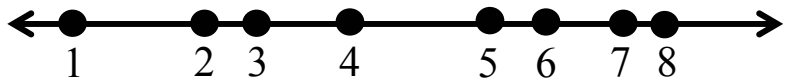
- Balance between
  - $P(n)$ = preprocessing time
  - $S(n)$ = space requirement
  - $Q(n)$ = query time
- Theorem: $Q(n) \approx O\left(\frac{n \text{ polylog}(n)}{\sqrt[d]{S(n)}}\right)$

$1d$ Interval Query

- Preprocess points into Range Tree

- $P(n) = O(n \log n)$
- $S(n) = O(n)$
- $Q(n) = O(\log n)$

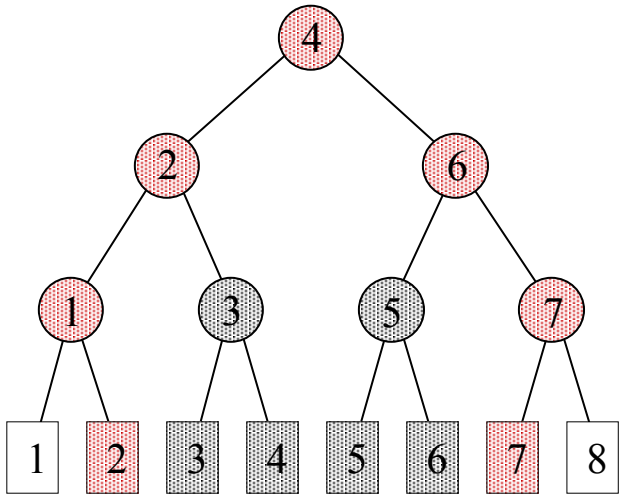


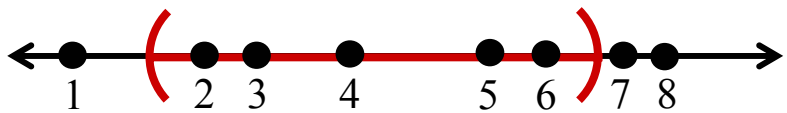


$1d$ Interval Query

- Preprocess points into Range Tree

- $P(n) = O(n \log n)$
- $S(n) = O(n)$
- $Q(t) = O(\log n)$





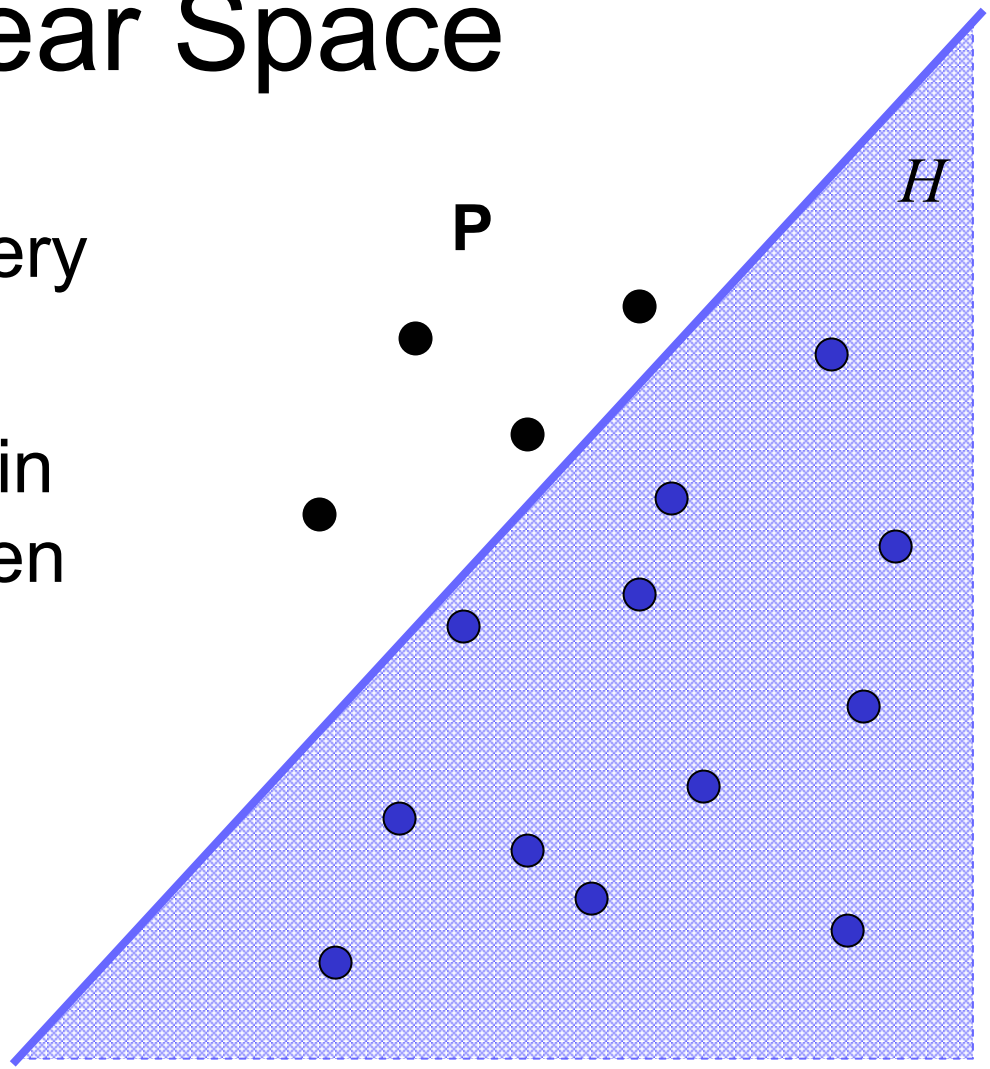
Extends to d-dim

- Orthogonal range queries – axis parallel
- Multi-level Range Tree
- Primary range tree = $1d$ tree on x-coord
- Canonical subsets for each internal node stored in secondary tree on y-coord, etc…

- $Q(n) = O(\log^d n)$, $S(n) = O(n \log^{d-1} n)$

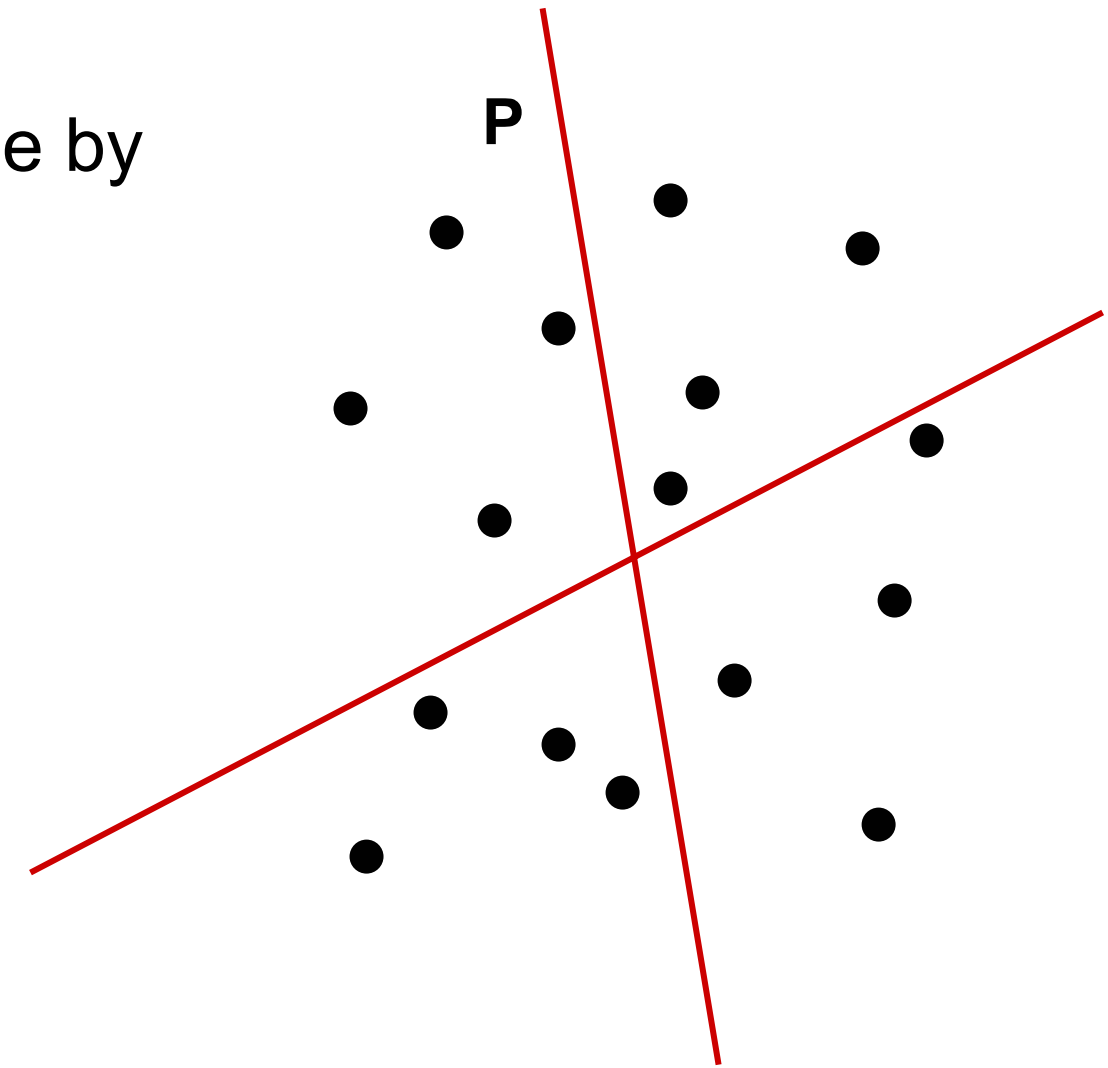
Linear Space

- $2d$ half-plane query
- Report all points in **P** that lie in a given half plane H
- Want $S(n) = O(n)$



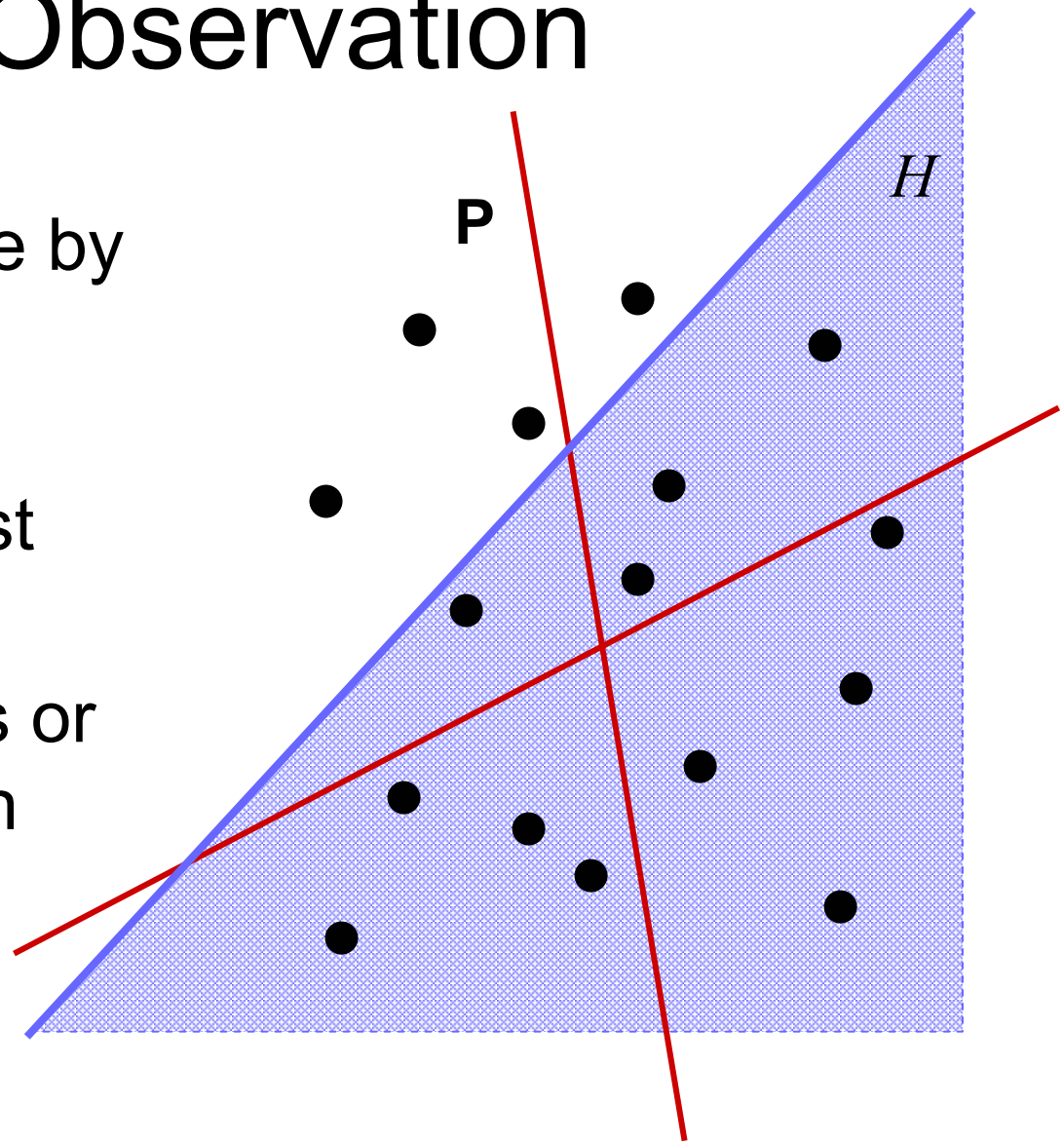
Key Observation

- If we divide plane by 2 lines, then:



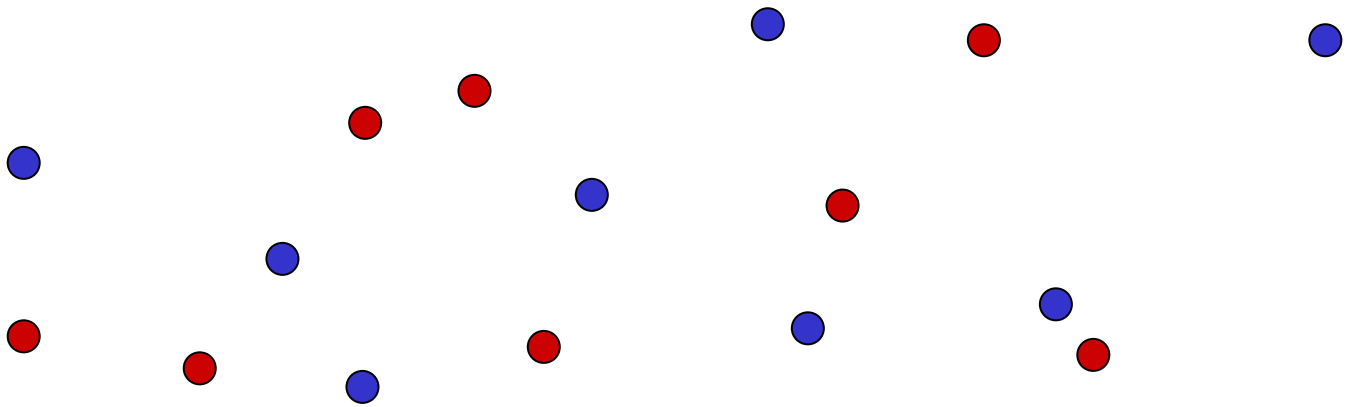
Key Observation

- If we divide plane by 2 lines, then:
- Boundary of H intersects at most 3 regions,
- H either contains or misses the fourth region entirely



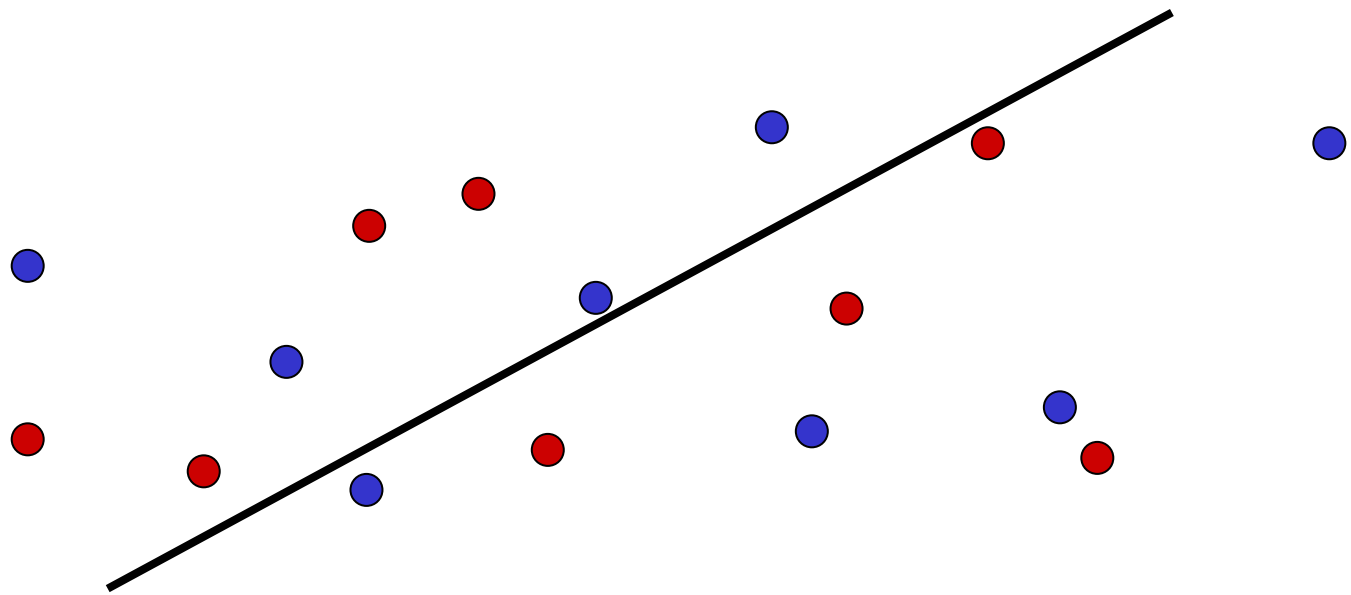
Ham-sandwich Theorem

- Given any two point sets P_1 and P_2 in the plane, it is always possible, in linear time, to find a line which bisects both of them.



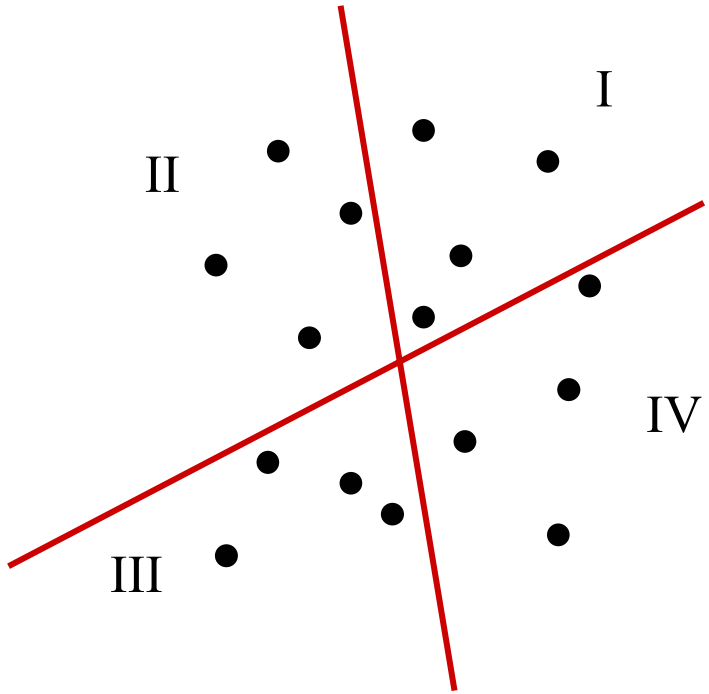
Ham-sandwich Theorem

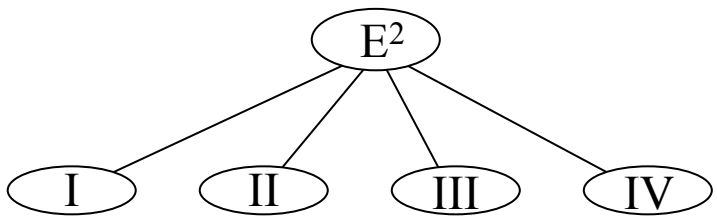
- Given any two point sets P_1 and P_2 in the plane, it is always possible, in linear time, to find a line which bisects both of them.



Partition Tree (Willard)

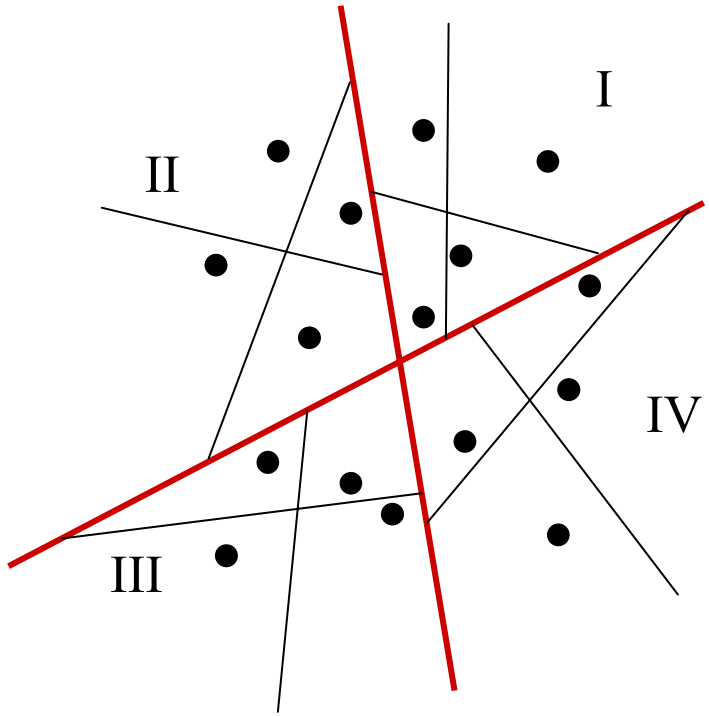
- Root = entire plane
- Choose 2 split lines to divide points evenly (by ham-sandwich)
- 4 children, one for each region

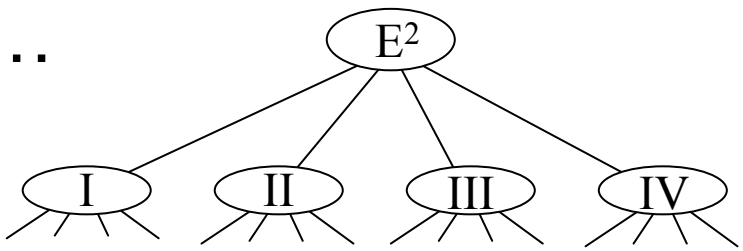




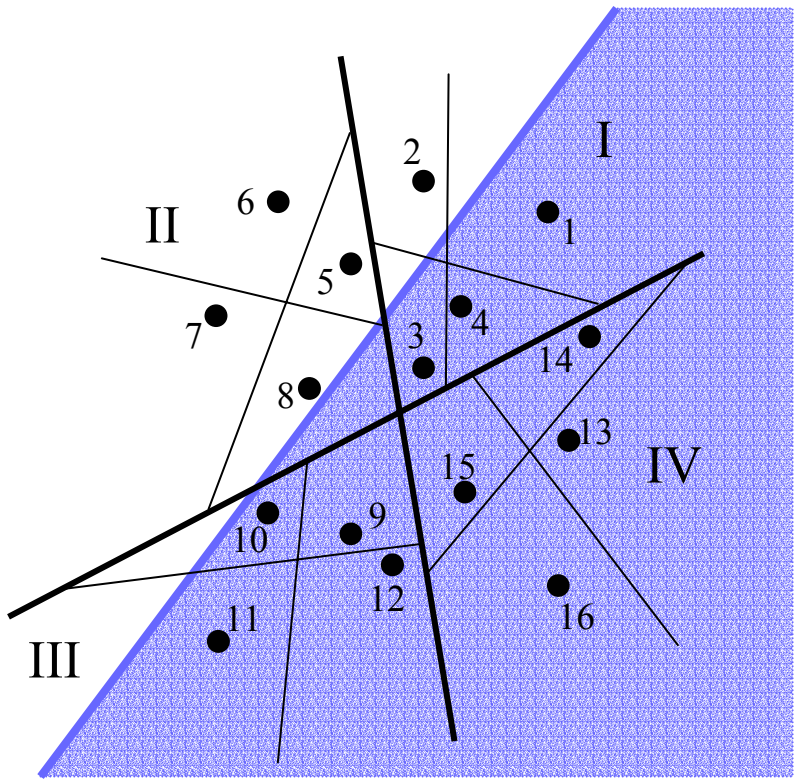
Partition Tree (Willard)

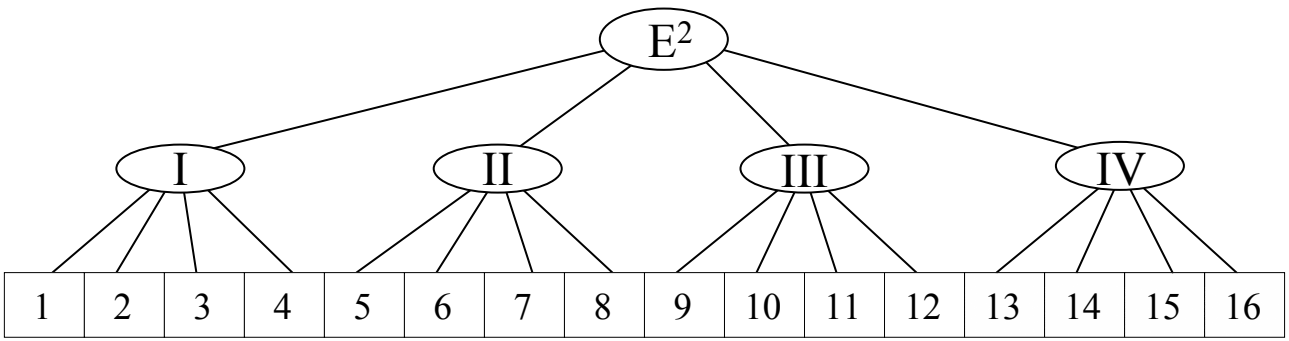
- Root = entire plane
- Choose 2 split lines to divide points evenly (by ham-sandwich)
- 4 children, one for each region
- Recurse…





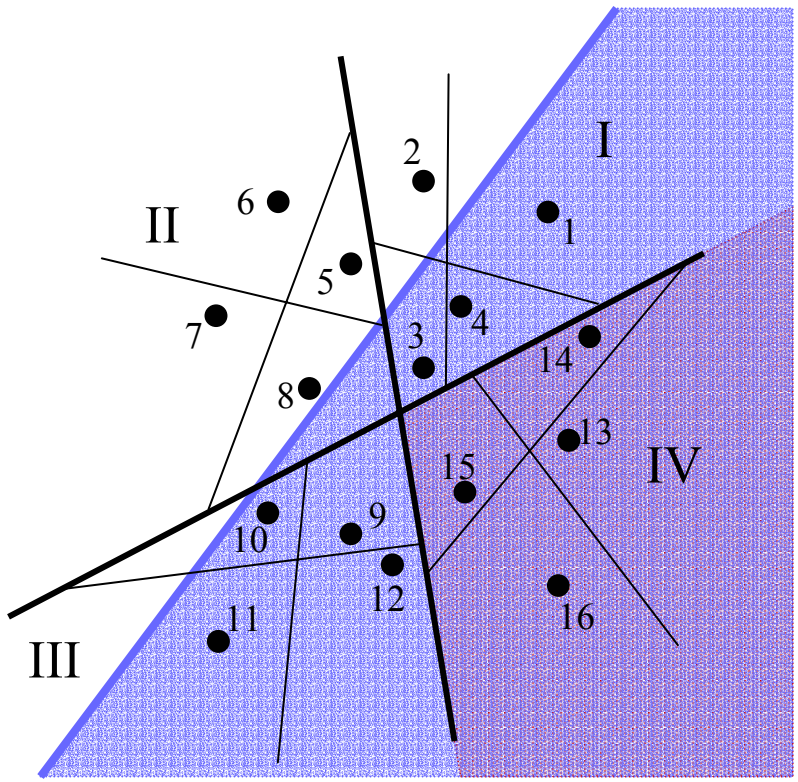
- Example:

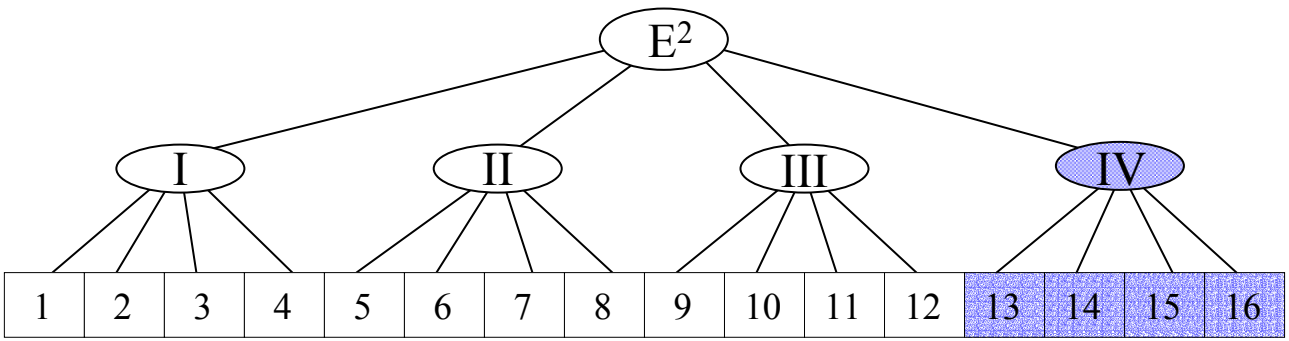

I
II
III
IV
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16


E2
I
II
III
IV
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

- Example:

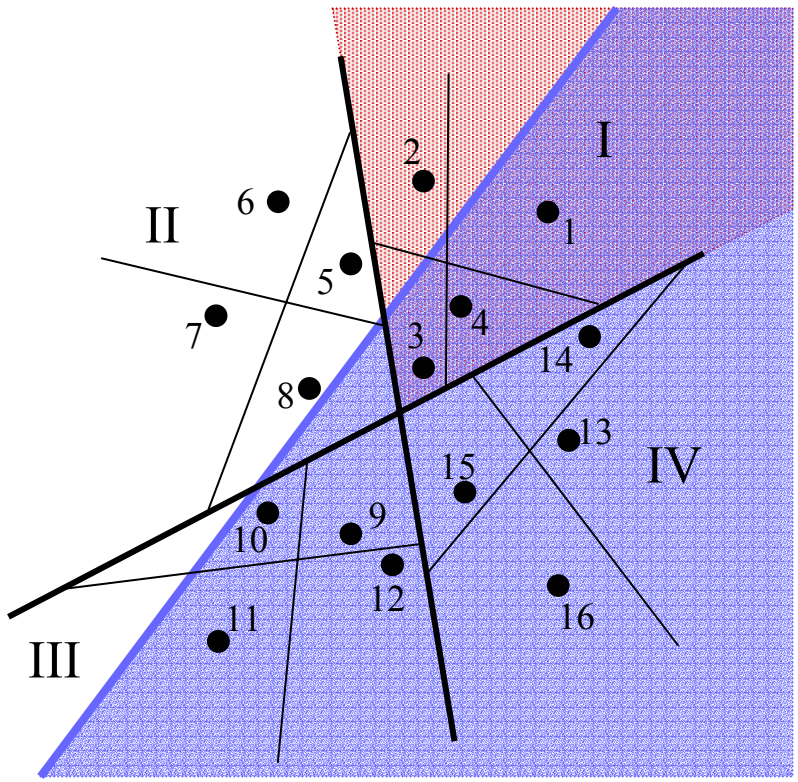
- H contains IV entirely

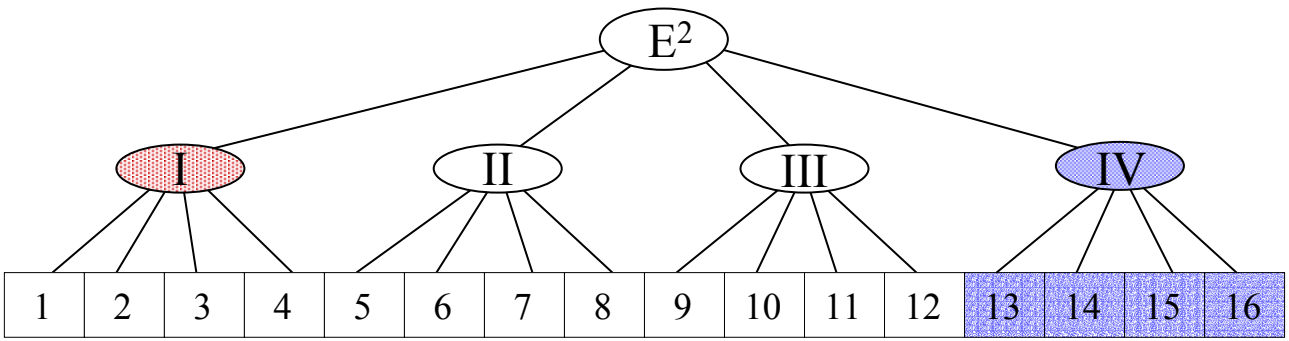




- Example:

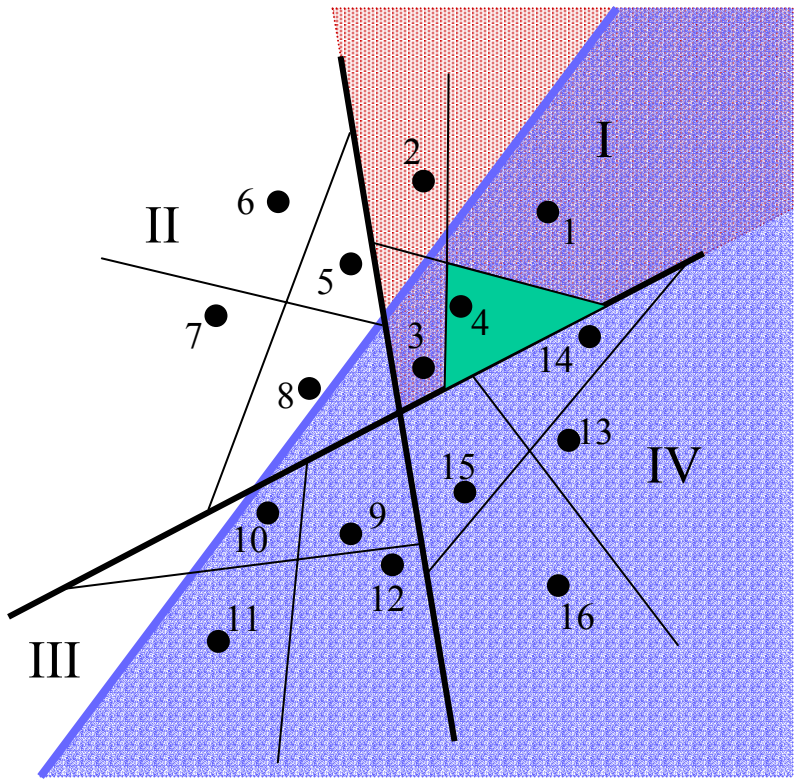
- H contains IV entirely
- Recurse on I

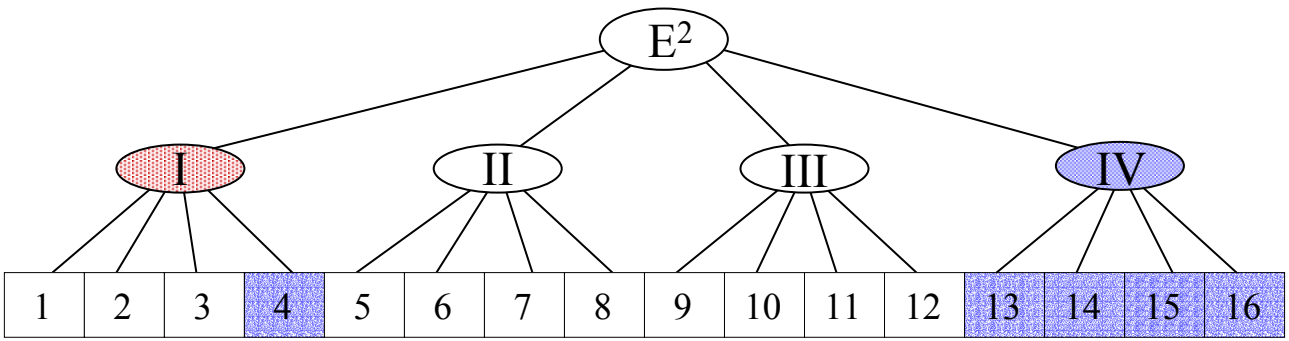




- Example:

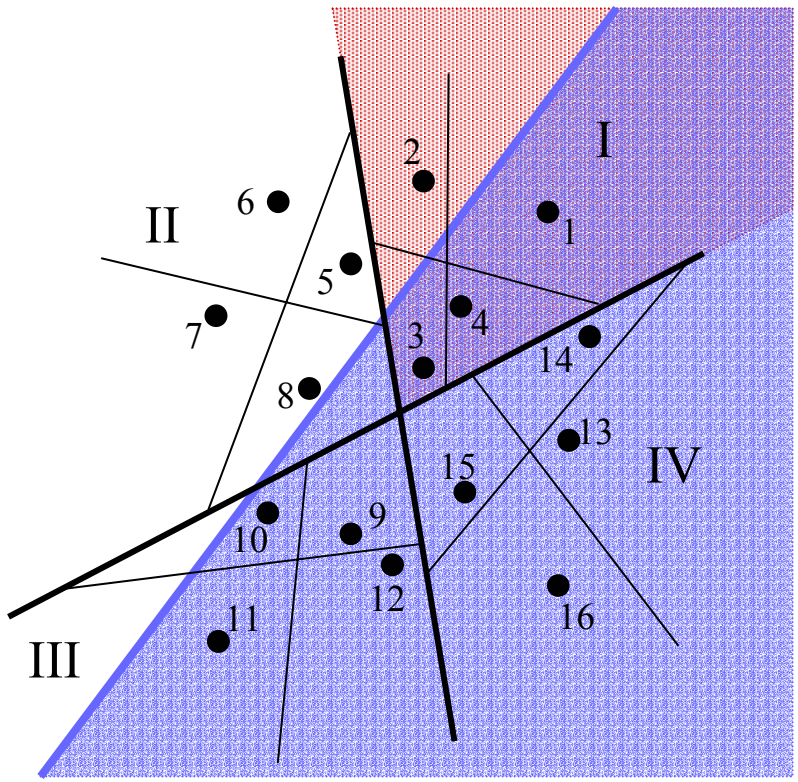
- H contains IV entirely
- Recurse on I
  - H contains 4

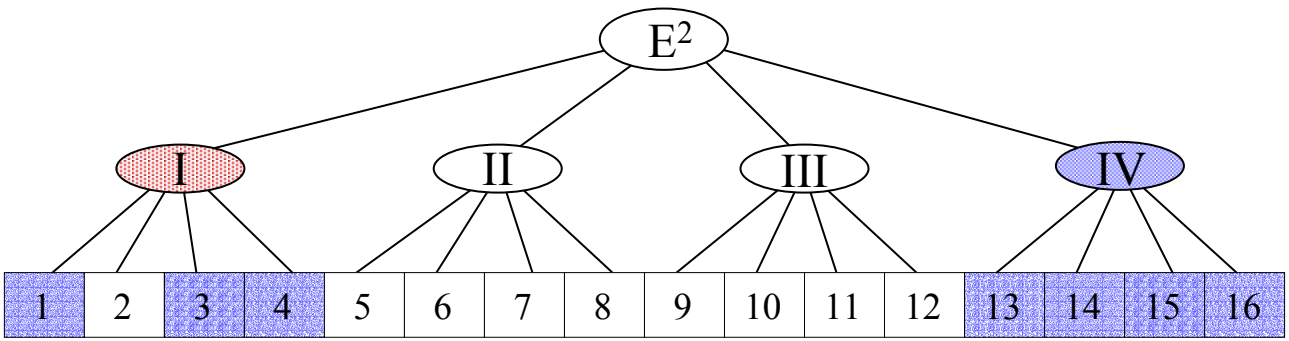




- Example:

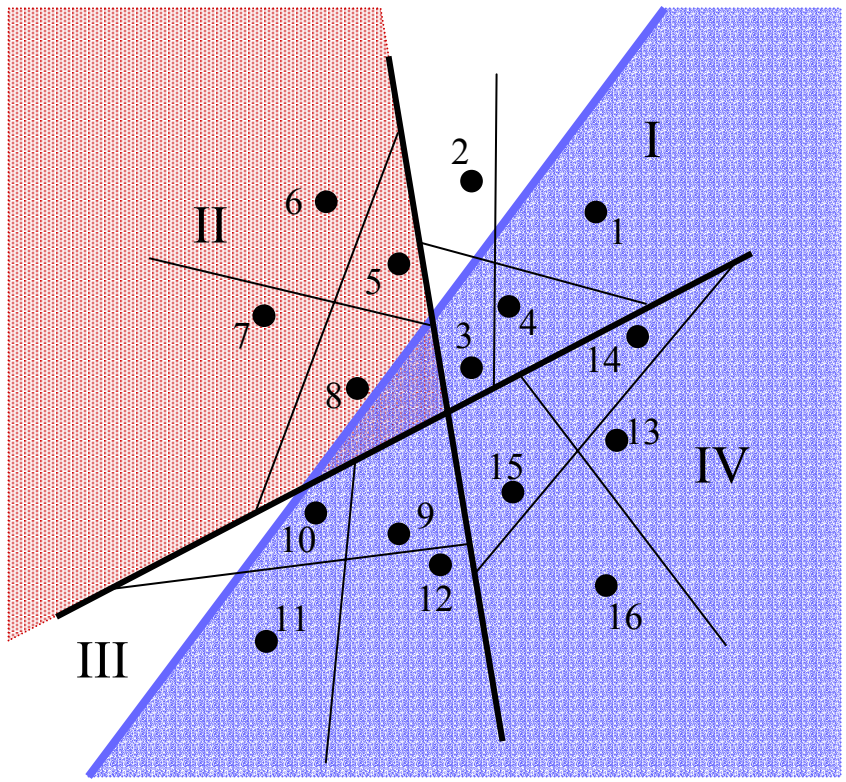
- H contains IV entirely
- Recurse on I
  - H contains 4
  - Recurse on 1,2,3 (by inspection)

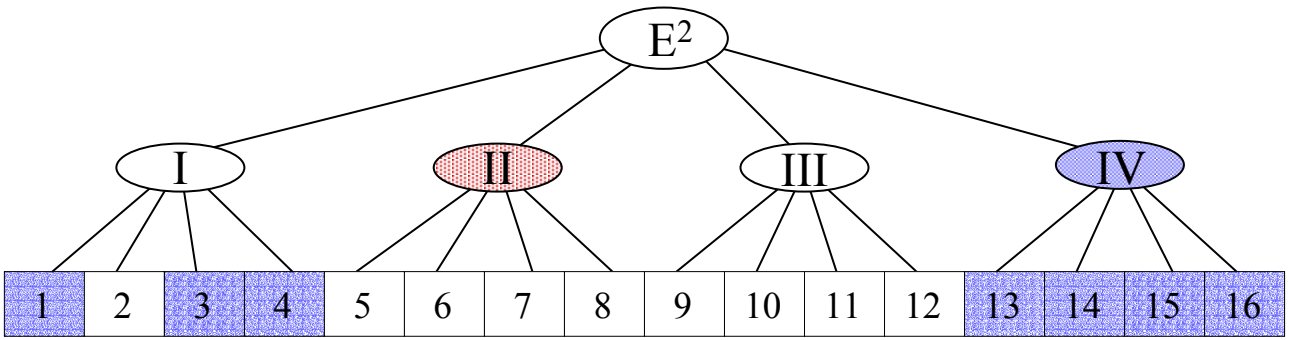




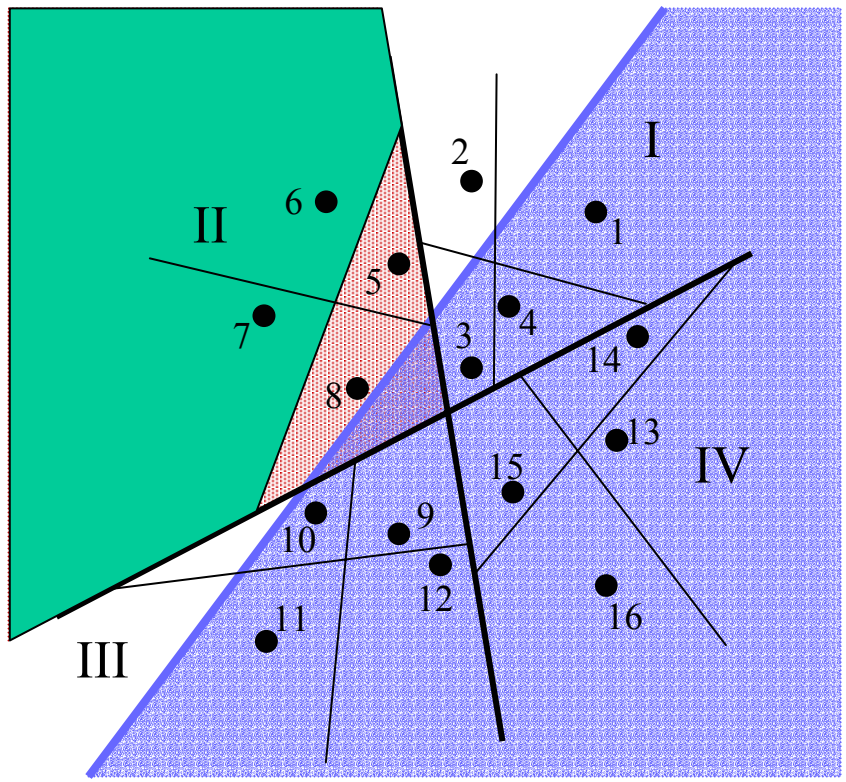
- Example:

- H contains IV entirely
- Recurse on II

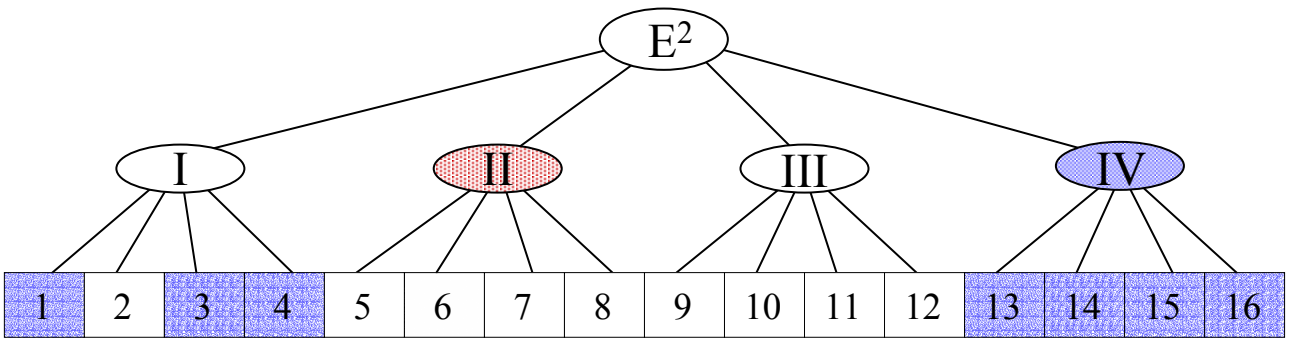




- Example:

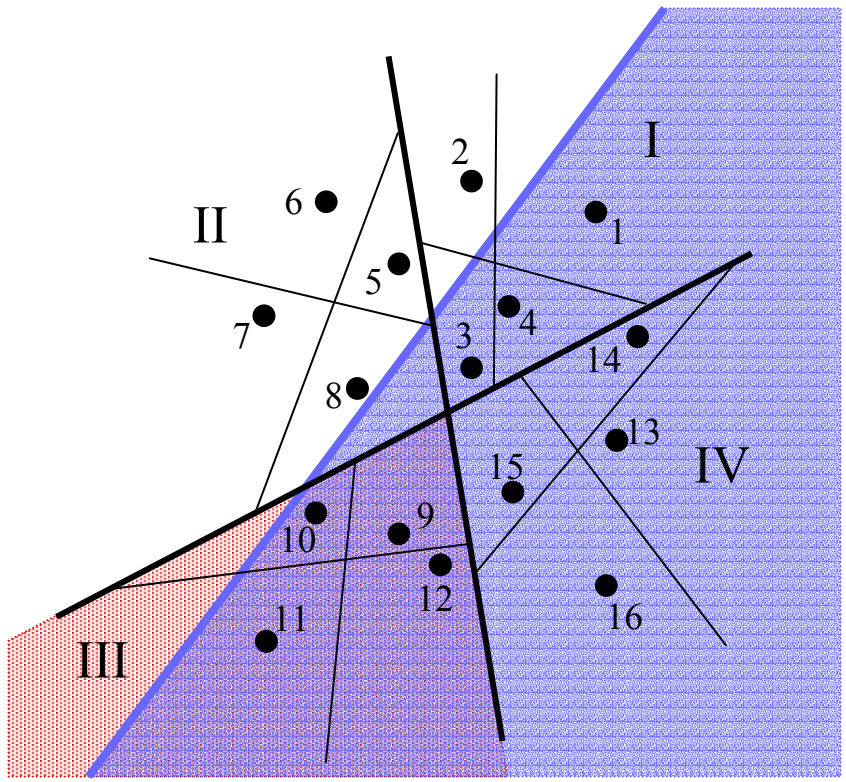


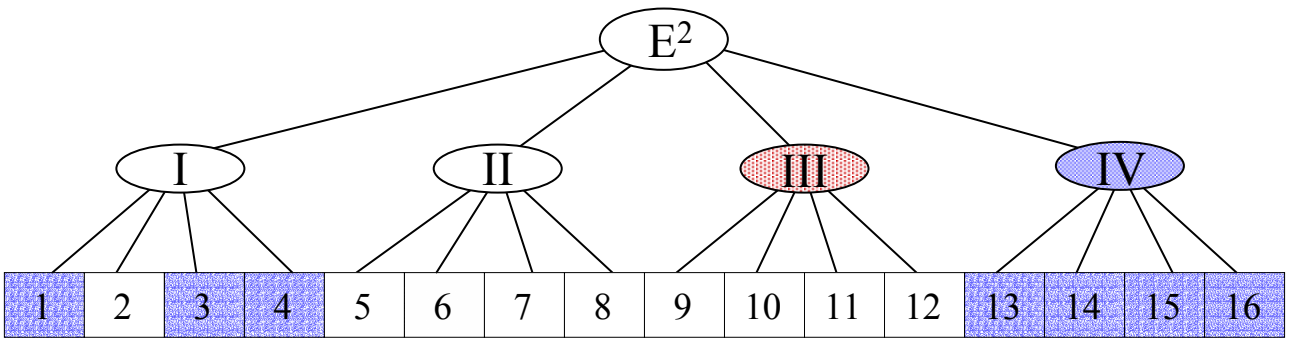
- H contains IV entirely
- Recurse on II
  - 6, 7 outside of H
  - Recurse on 5,8 (by inspection)



- Example:

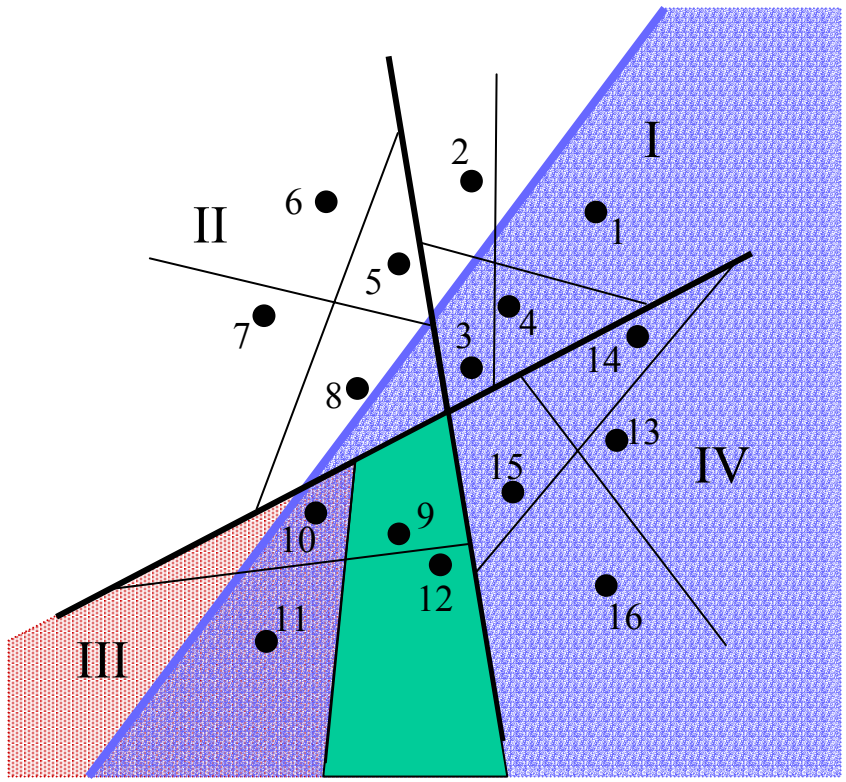
- H contains IV entirely
- Recurse on III

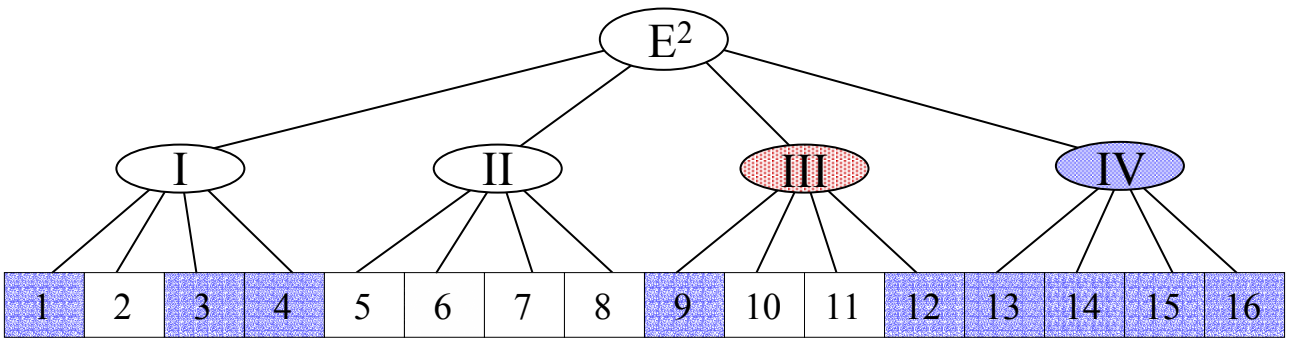




- Example:

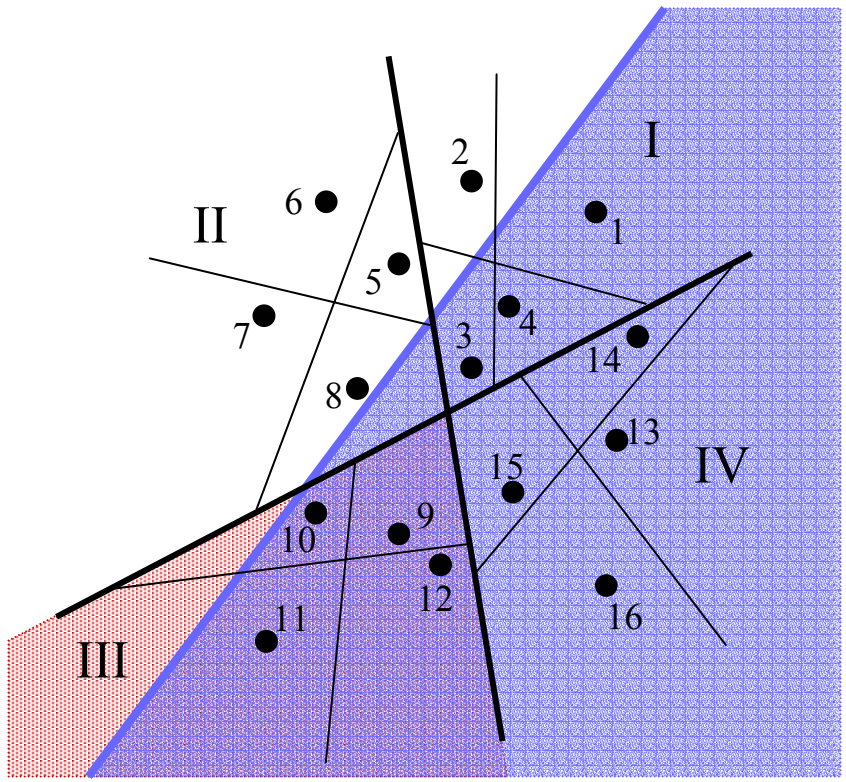
- H contains IV entirely
- Recurse on III
  - H contains 9, 12
  - Recurse on 10, 11 (by inspection)

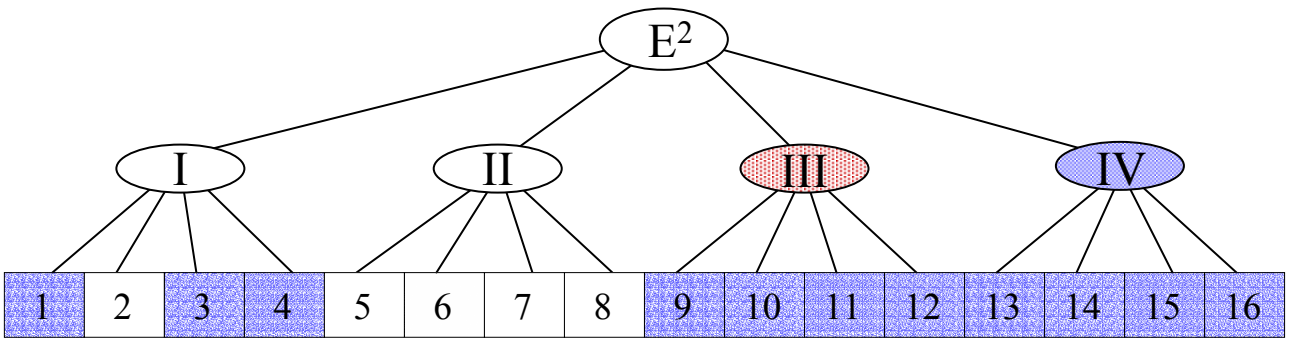




- Example:

- H contains IV entirely
- Recurse on III

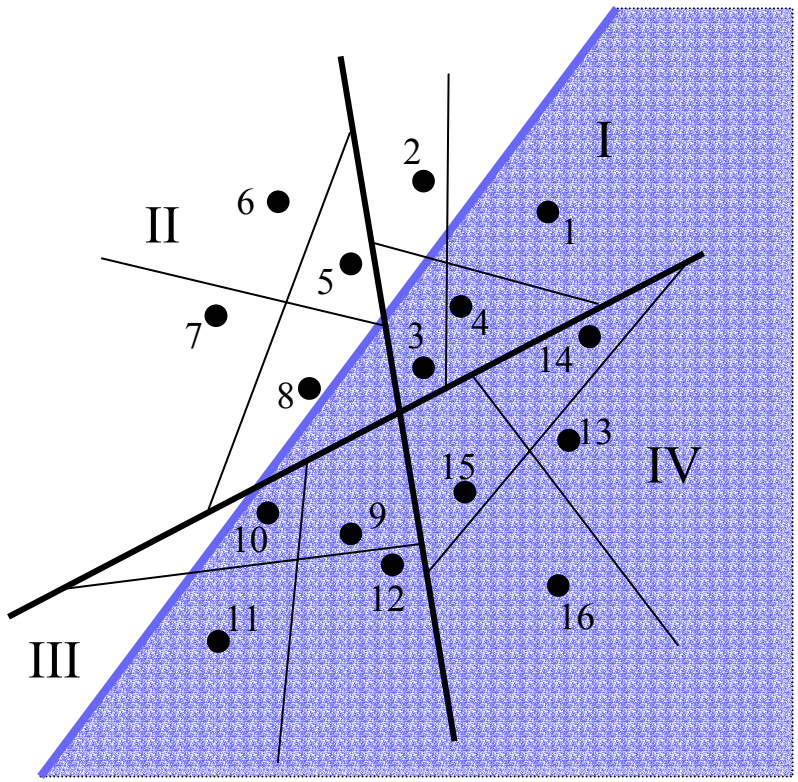


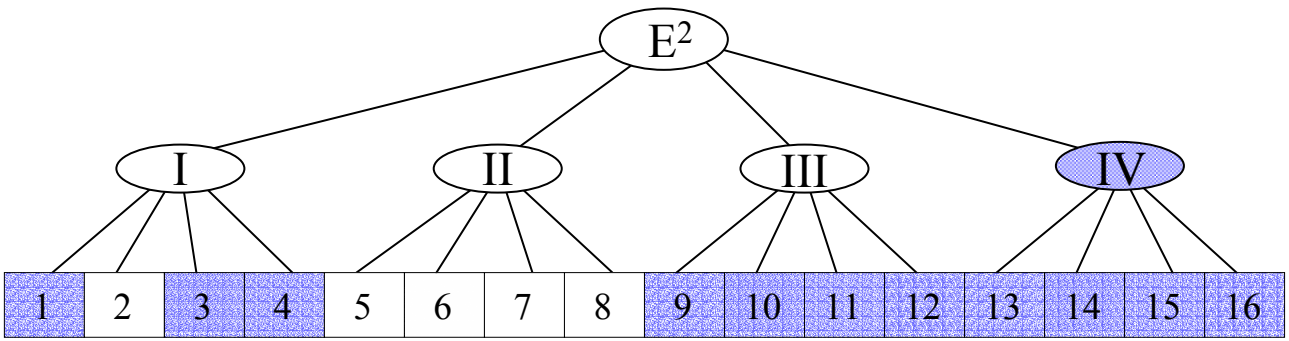


- Example:

- H contains IV entirely
- Recurse on III

- H contains 1,3,4,9-16





Performance of Willard Tree

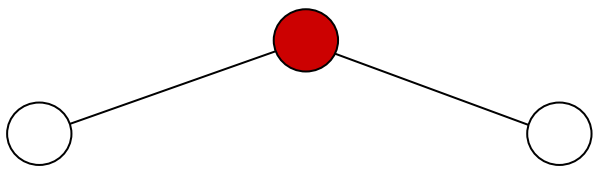
- $S(n) = \mathrm{O}(n)$
  - Tree with n leaves

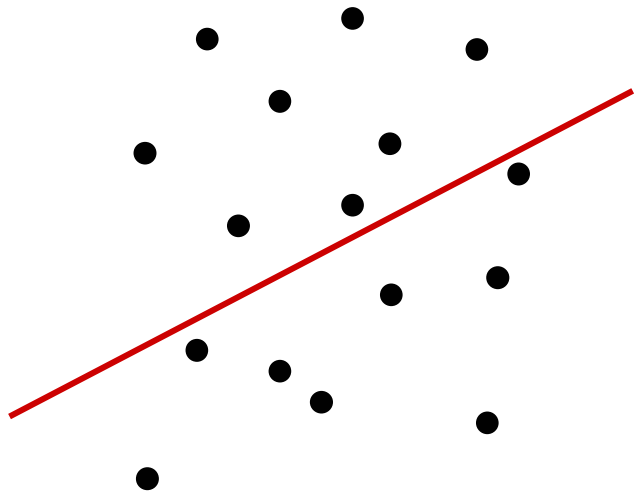
- $P(n) = O(n \log n)$
  - ham-sandwich is $O(n)$, so $O(n)$ for each of $\log n$ levels

- $Q(n) \;=\; 3Q(n/4) + O(1) \;=\; O(n^{\log_4 3}) \;= O(n^{0.792})$

Partition Tree II (Edelsbrunner)

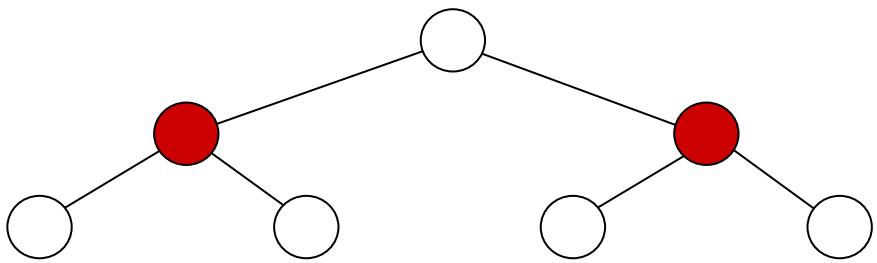
- Binary tree

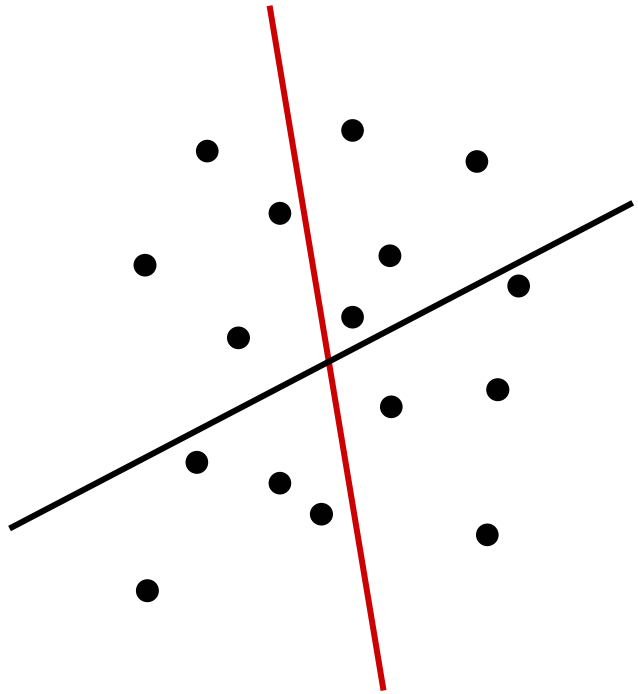




Partition Tree II (Edelsbrunner)

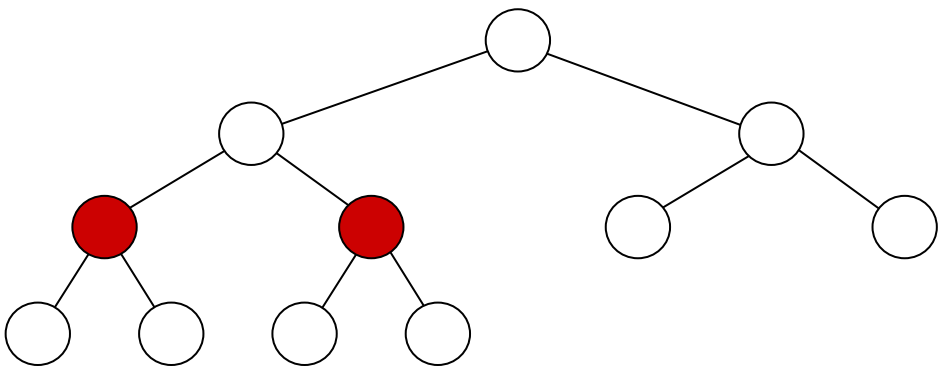
- Binary tree
- Cut line bisects both children

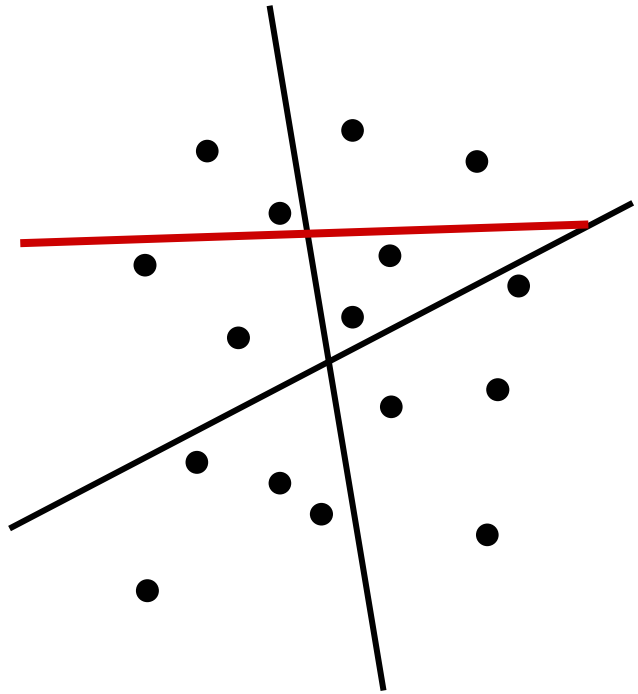




Partition Tree II (Edelsbrunner)

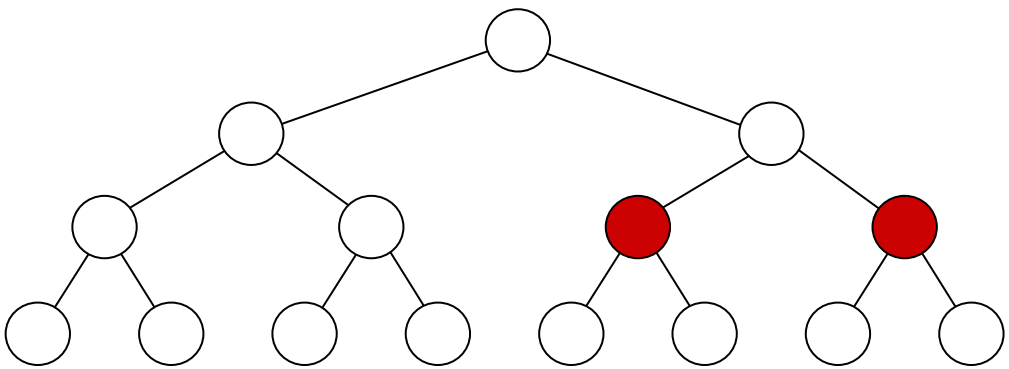
- Binary tree
- Cut line bisects both children

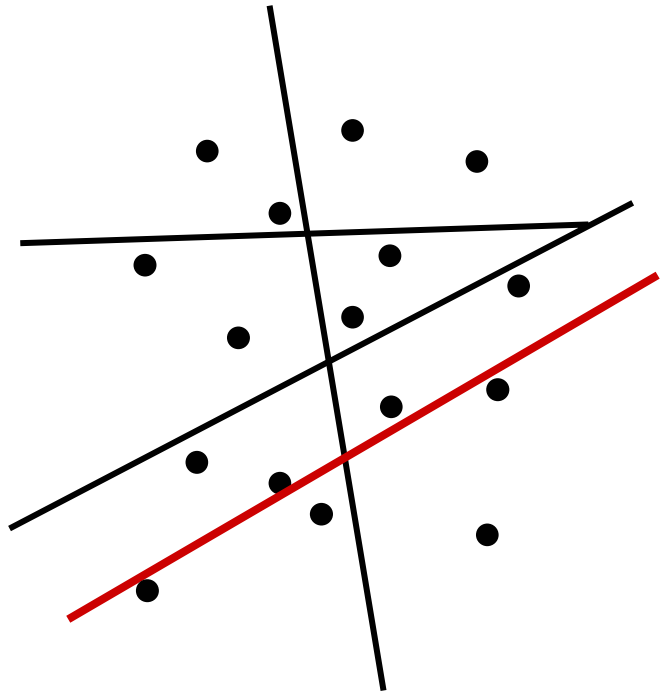




Partition Tree II (Edelsbrunner)

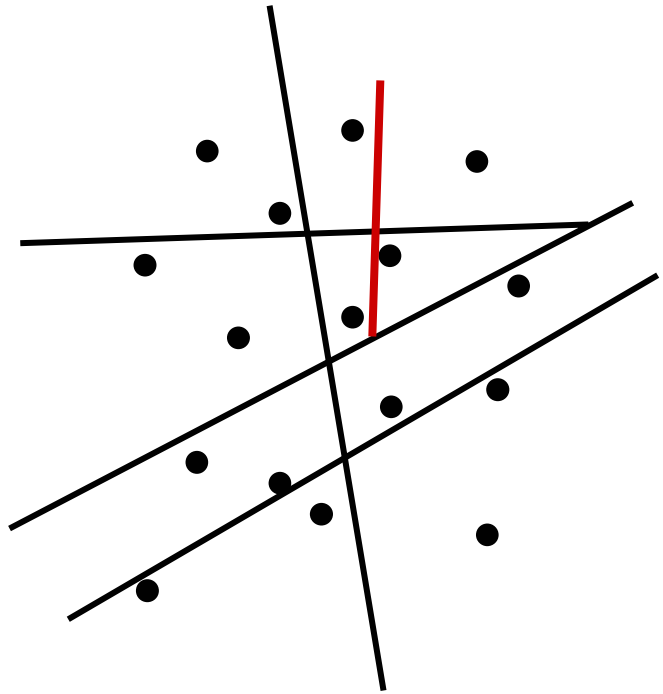
- Binary tree
- Cut line bisects both children

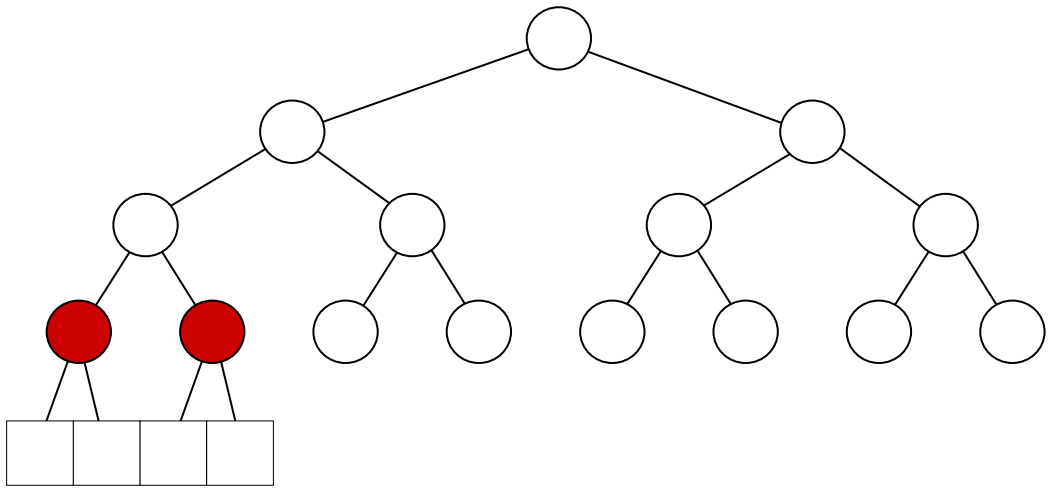




Partition Tree II (Edelsbrunner)

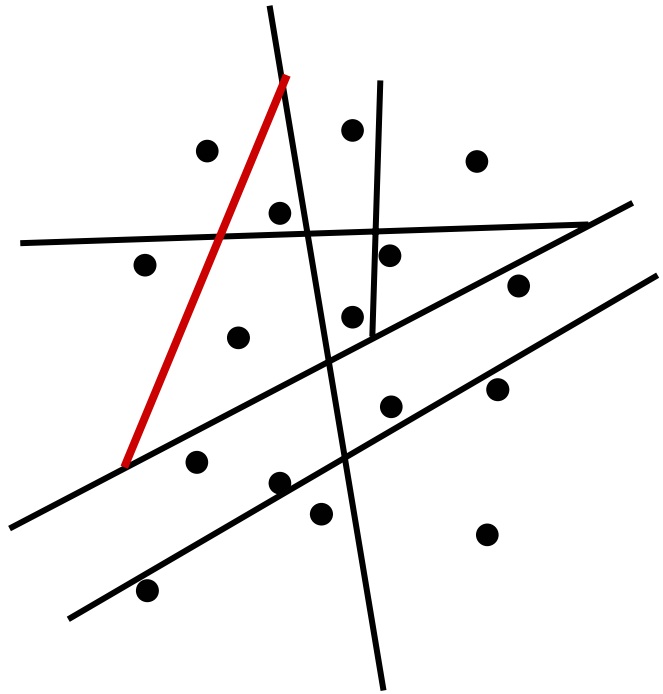
- Binary tree
- Cut line bisects both children

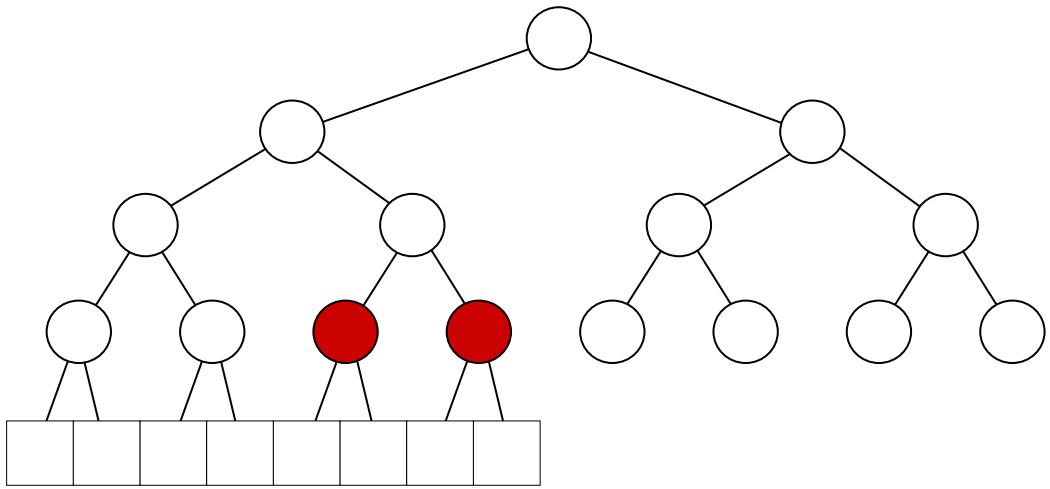




Partition Tree II (Edelsbrunner)

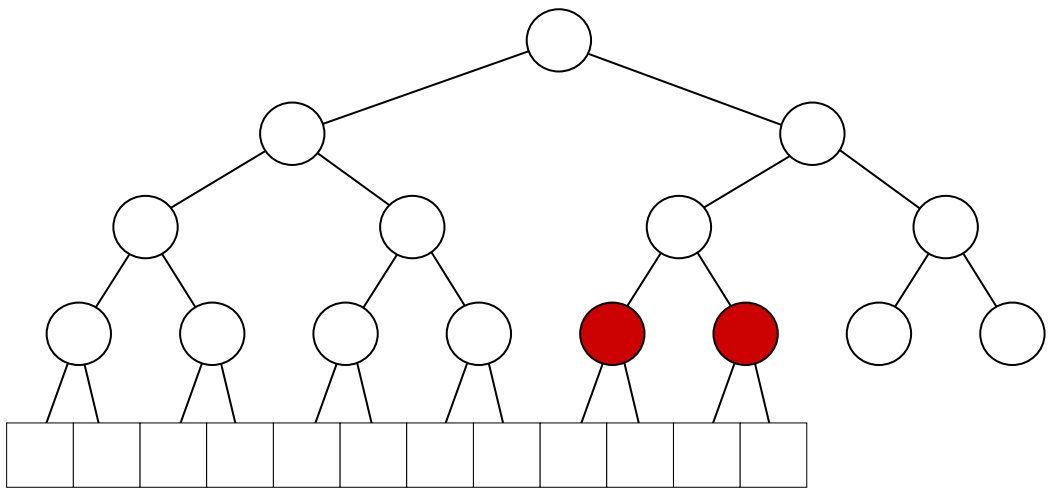
- Binary tree
- Cut line bisects both children

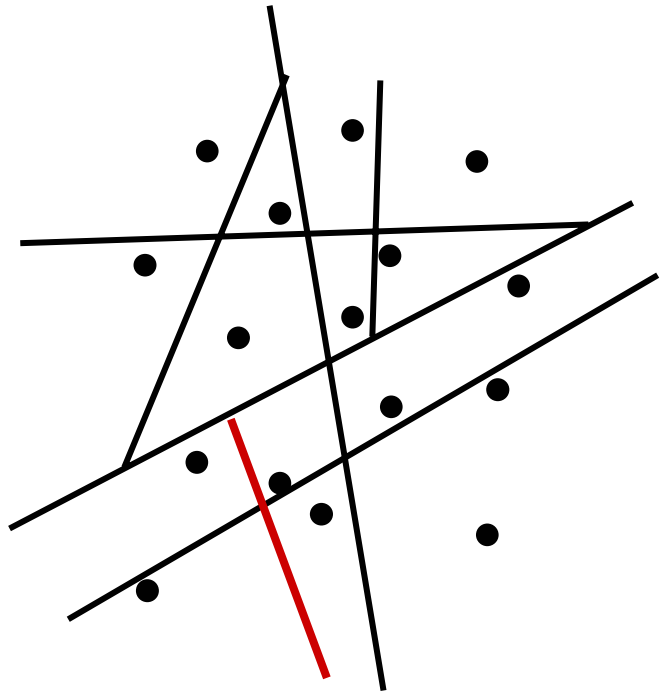




Partition Tree II (Edelsbrunner)

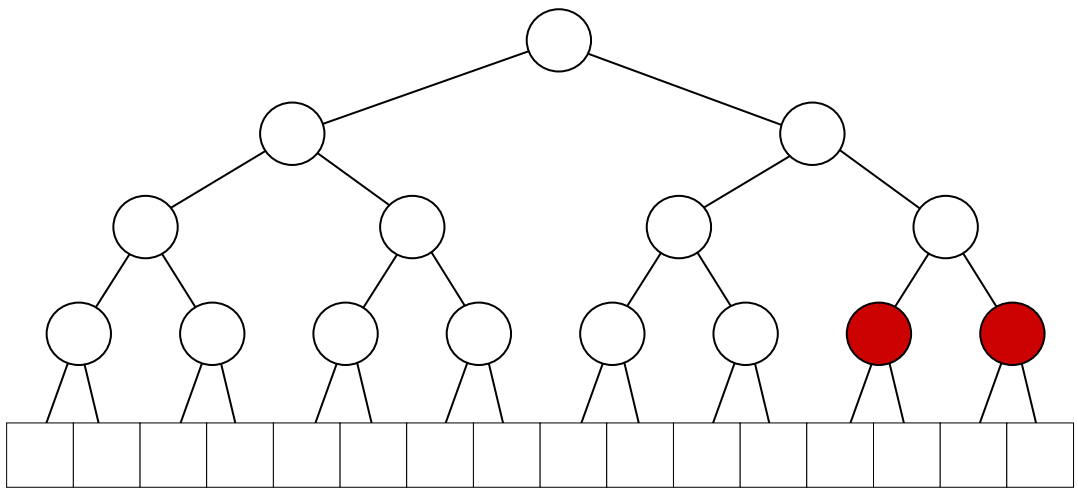
- Binary tree
- Cut line bisects both children

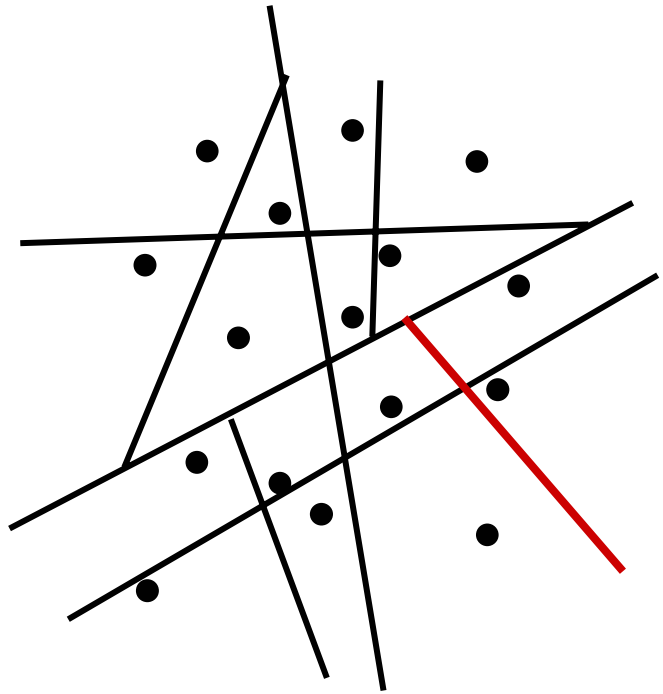




Partition Tree II (Edelsbrunner)

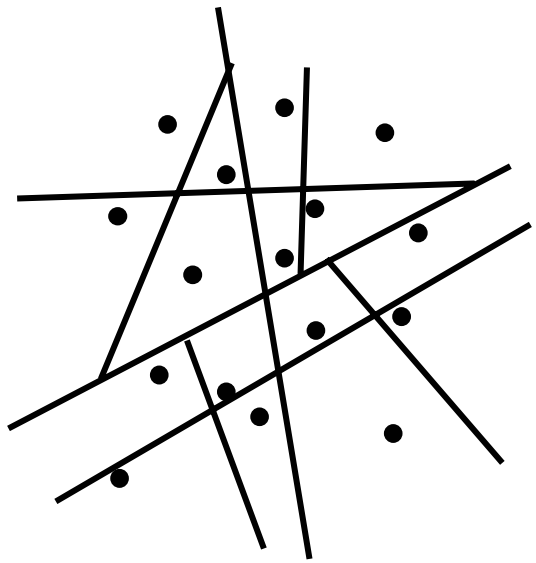
- Binary tree
- Cut line bisects both children

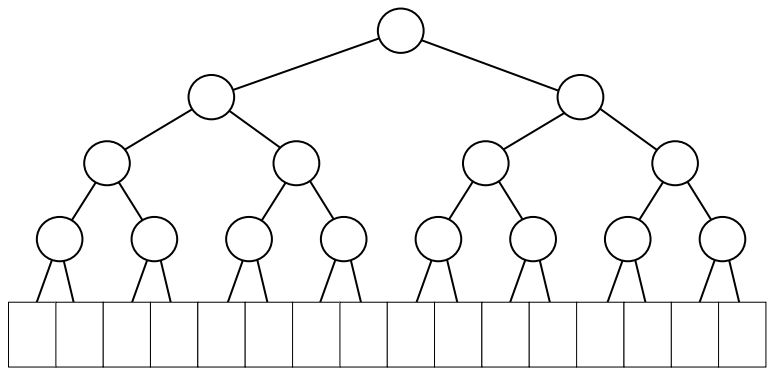




Partition Tree II (Edelsbrunner)

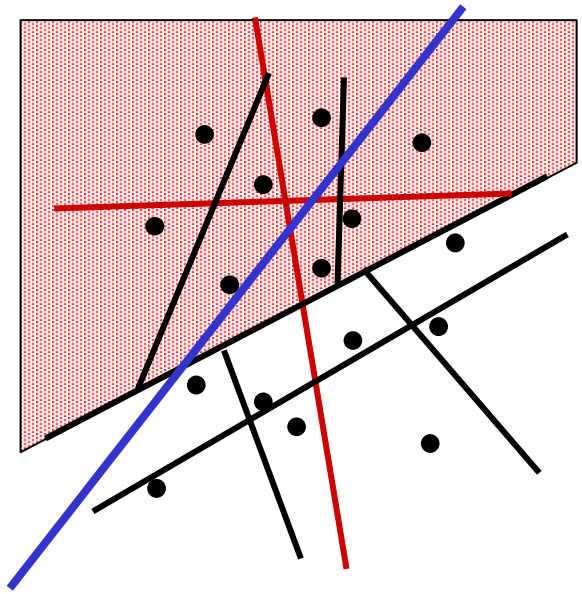
- $S(n) = O(n)$
- $P(n) = O(n \log n)$
- $Q(n) = ?$

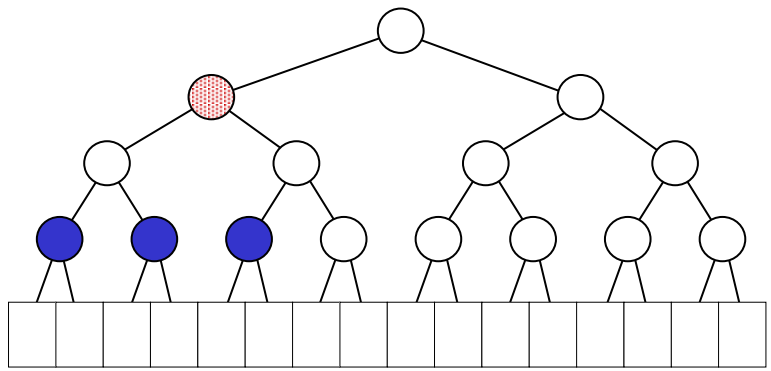




Partition Tree II (Edelsbrunner)

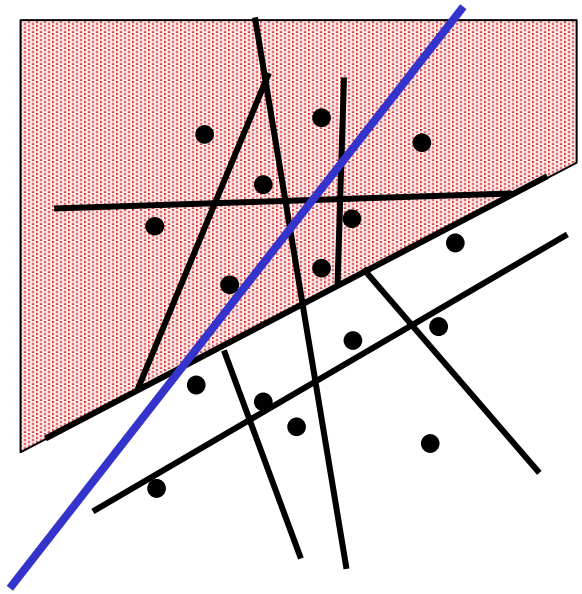
- $S(n) = O(n)$
- $P(n) = O(n \log n)$
- $Q(n) = ?$





Partition Tree II (Edelsbrunner)

- $S(n) = O(n)$
- $P(n) = O(n \log n)$
- $Q(n) =$
  $Q(n/2) + Q(n/4) + O(1)$
  $= O(n^{0.695})$



- Closer to optimal $O(n^{1/2})$