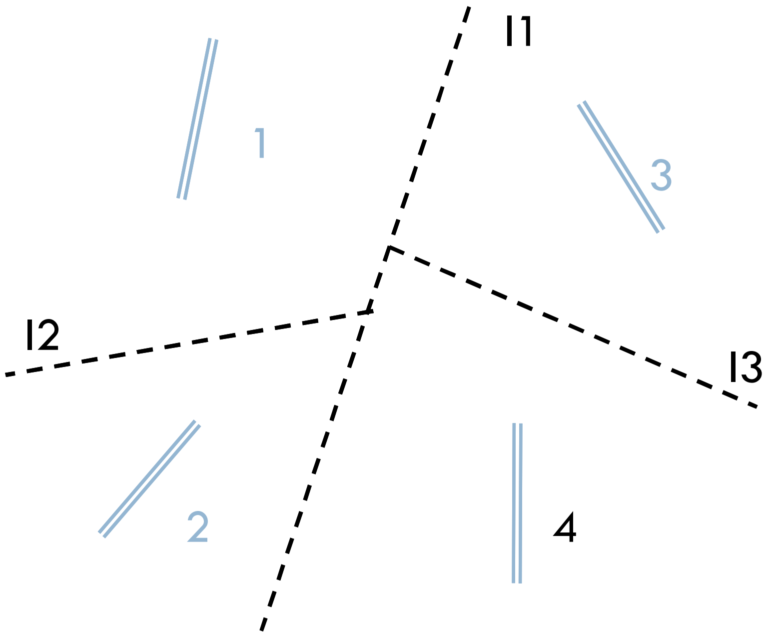
BINARY PLANAR PARTITION

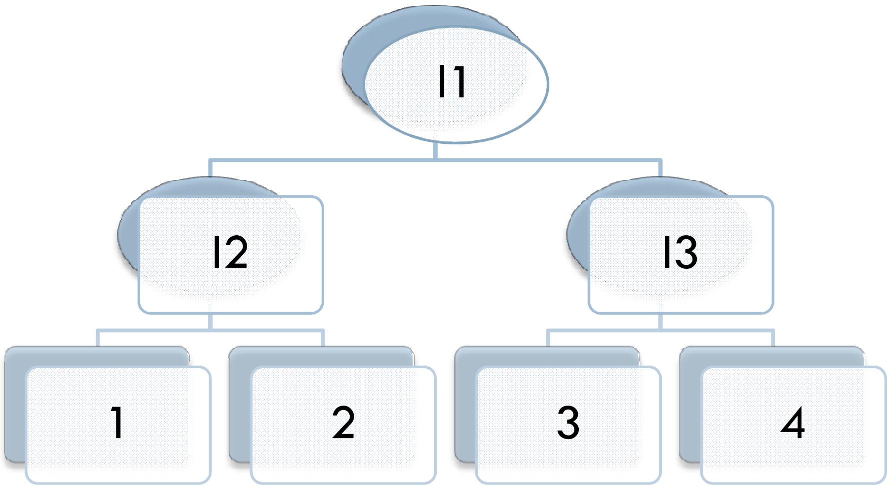
Lecture 2

Advanced Algorithms II

Slides by: Arjun Dasgupta

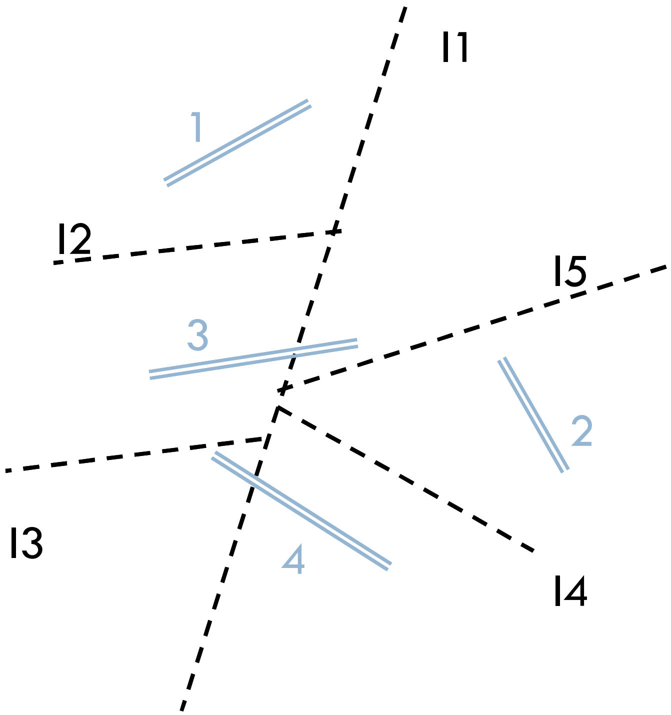
Example 1

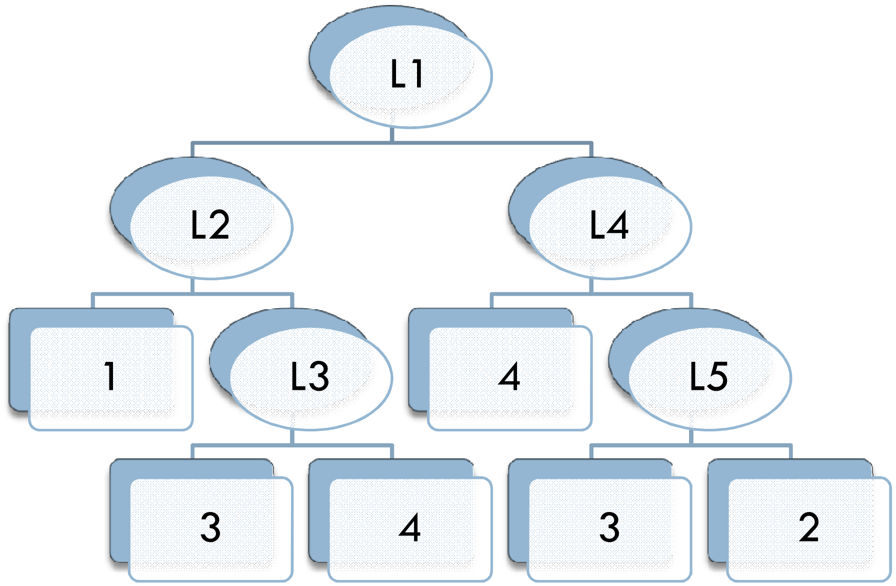




- Each oval node stores information about the infinite line l_i
- The leaves denote the line segments being partitioned

Example 2





- Smallest Tree that can be created from the partitions is ***O(n)***

Auto-Partition Algorithm

- Index(u , v) = # of cuts that *u* makes when extended to *v*

Algorithm:

Input: $S = \{S_1, S_2, \dots S_n\}$

1. Generate a random permutation of *S*
   $U = \{u_1, u_2, \dots u_n\}$
2. Start constructing the tree by using the segments in this order as partitioning lines

✓ Upper Bound of the size of tree created by Auto-Partition -> $O(n)$

Analysis

- Our objective is to calculate

  $\sum^{n}_{i=1}\sum^{n}_{j\neq i, j=1}$ Prob(i cuts j)
- Now,

  $\sum^{n}_{j\neq i}$ Prob(i cuts j) $\leq (1/2 + 1/3 + \ldots..)$

  $\leq 2 \, ln \, n$

  And, $\sum^{n}_{i=1}\sum^{n}_{j\neq i, j=1}$ Prob(i cuts j) $\leq 2 \, n \, ln \, n$
- Thus,

  E[# of cuts] $\leq 2 \, n \, ln \, n$ and,

  E[Tree Size] = $O(nlogn)$