BINARY PLANAR PARTITION
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

- $\text{Index}(u,v) = \# \text{ of cuts that } u \text{ makes when extended to } v$

**Algorithm:**

Input: $S = \{S_1, S_2, \ldots, S_n\}$
1. Generate a random permutation of $S$
   
   $U = \{u_1, u_2, \ldots, u_n\}$
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 $\rightarrow O(n)$
Analysis

- Our objective is to calculate
  \[ \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} \text{Prob}(i \text{ cuts } j) \]

- Now,
  \[ \sum_{j \neq i}^{n} \text{Prob}(i \text{ cuts } j) \leq (1/2 + 1/3 + \ldots) \]
  \[ \leq 2 \ln n \]

- And,
  \[ \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} \text{Prob}(i \text{ cuts } j) \leq 2 n \ln n \]

- Thus,
  \[ E[\# \text{ of cuts}] \leq 2 n \ln n \text{ and,} \]
  \[ E[\text{Tree Size}] = \Omega(n \log n) \]