LEC9 Graph Algorithm

3.26 https://github.com/GUMI-21/MIT6.006_note

Terminology

```
G = (V,E)
V = Vertices
E = Edges in VxV
```

Simple Graph

- No self loops
- Every edge is distinct
- $|E| = O(|V|^2)$
  Directed:
  |E| <= 2 (|V| 2) (V choose 2) = O(|V|^2)
  Undirected
  |E| <= (|V| 2) (V choose 2) = O(|V|^2)
- *neighbors*
  Adjecent vertices
- *degree*

Adjaceny list

Set maps vertex v as Adj(v).
May Store Adj(v) as direct access array hash table.

Path

Model Graph Problems

- Single_pair_reachability(G, s, t):
  is there a path in G from s to t ?
- Single_pair_shortest_path(G, s, t):
  return distance from s to t and a shortest path
- single_source_shortest_paths(G, s):
  Return shortest distance from s to all t plus a shortest path tree

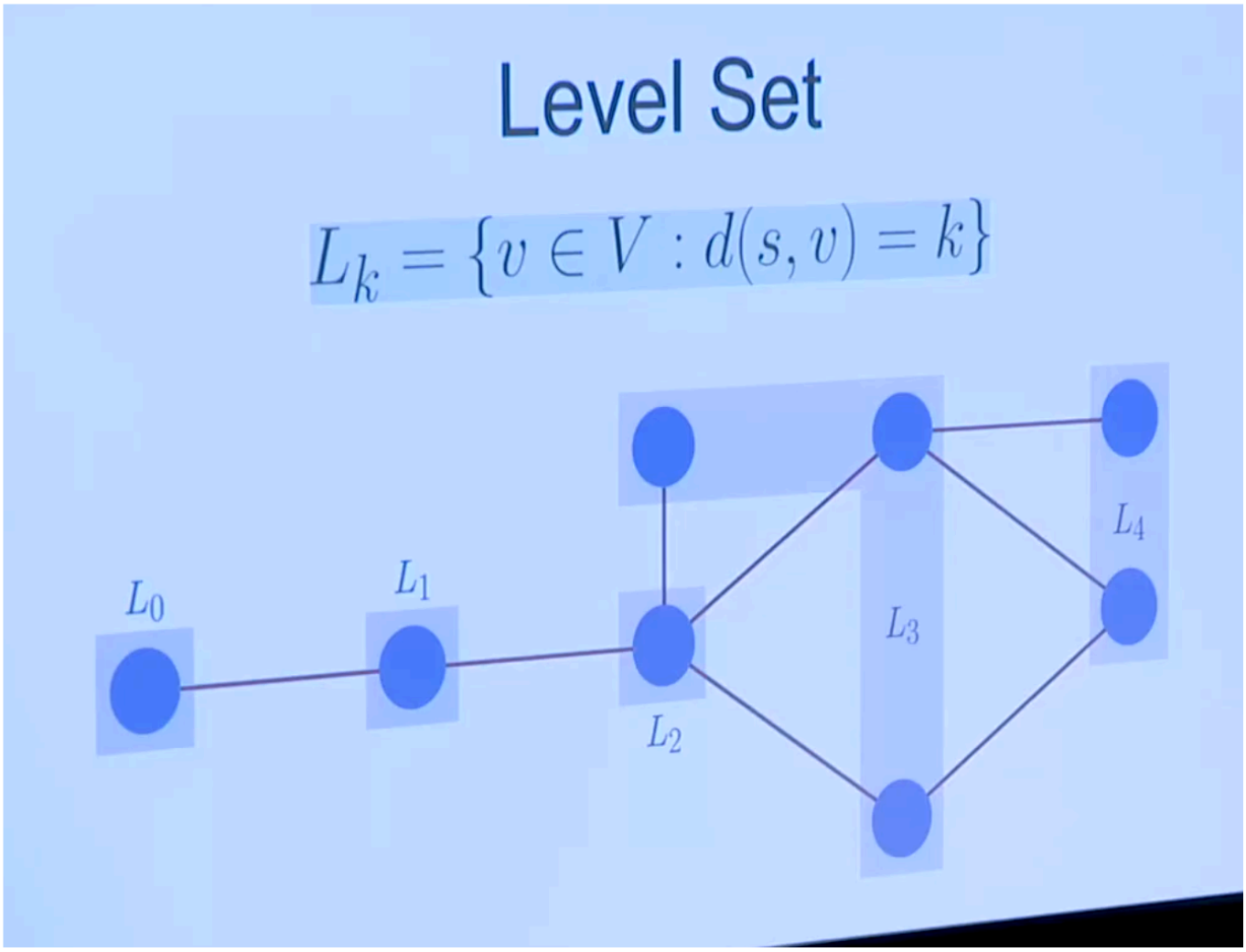
The Sortest path tree

every vertix just store one thing which is the previous vertex on its shotest path.

- P(V)
  previous of V
  but if we change the source or one edge, we may need to renew every P(V) in the grap.

Alorithm

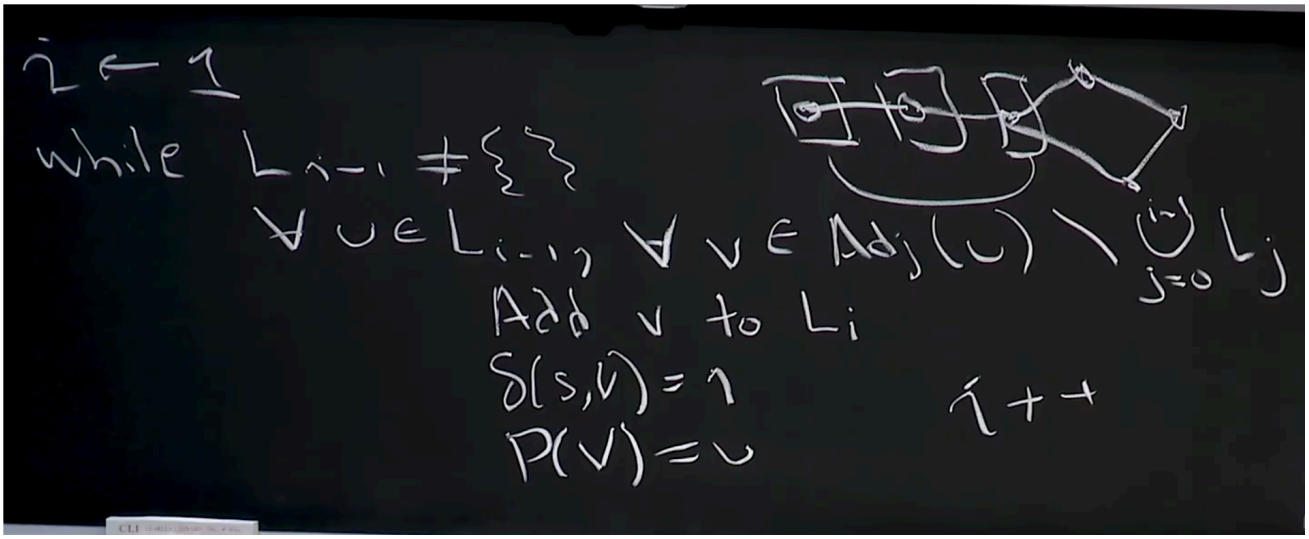
Level Set



Breadth-First Search

- init
  L_0 {s}
  P array {} *inital vertix lenth array O(|V|)*
  level set {[L_0]}

- then
  i = 1
  while L_{i-1} != {}



  while loop: we going with the order of distance, means take O(|E|) time.
  so the runtime is O(|E| + |V|)