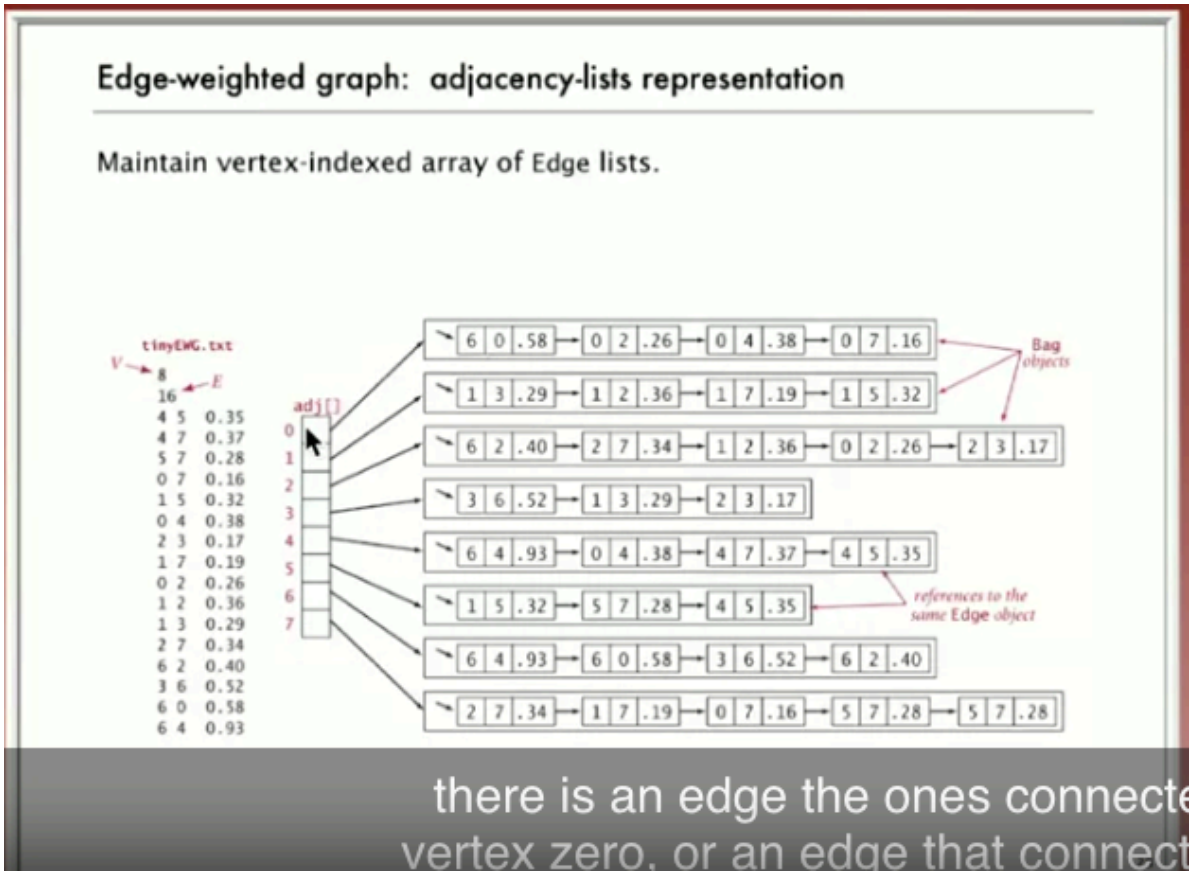
Graph Studies

Graph API

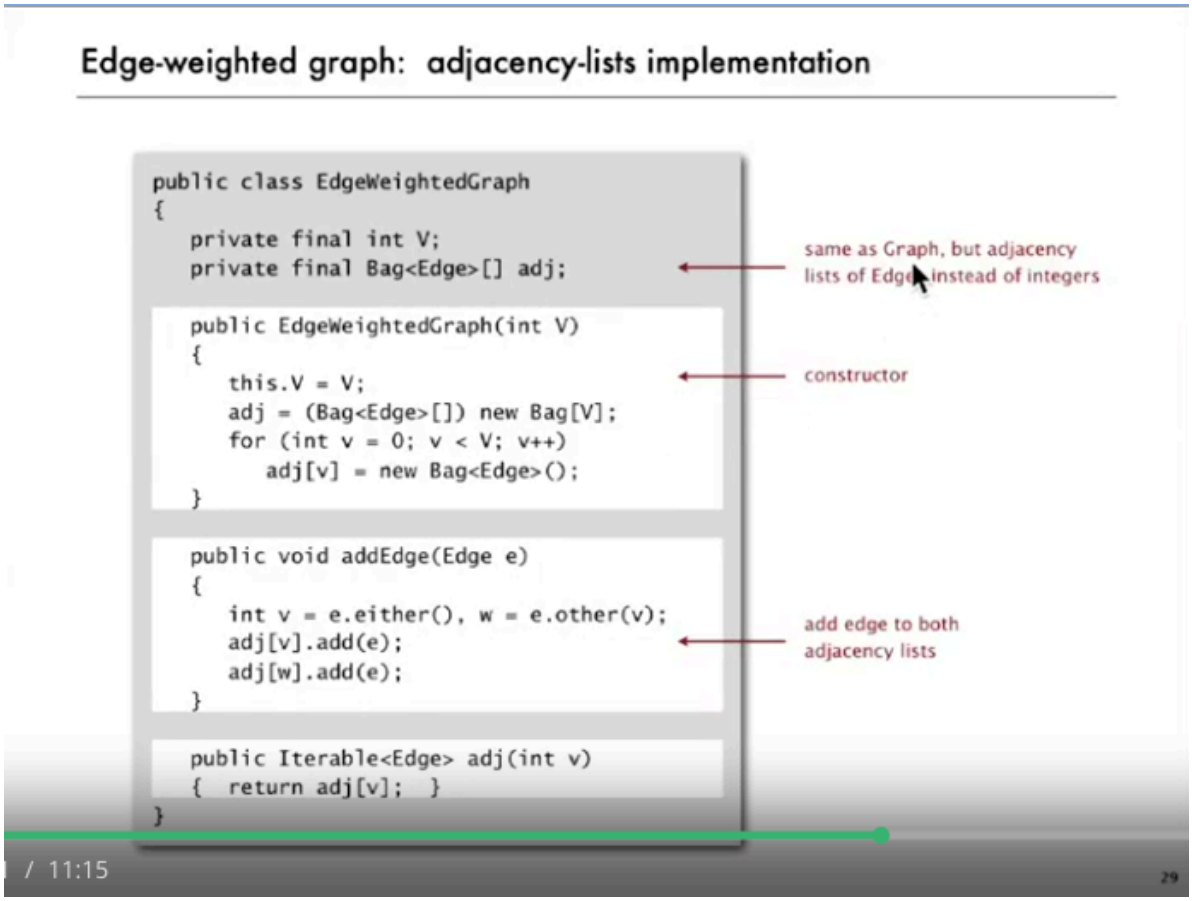
```
public class Edge implements Comparable<Edge>
        Edge(int v, int w, double weight)   //constructor
    int either()        //return the start point
    int other(int v)    //return other vertex
    int compareTo(Edge that) //比较权重
```

维护一个数组adj[EdgeList]



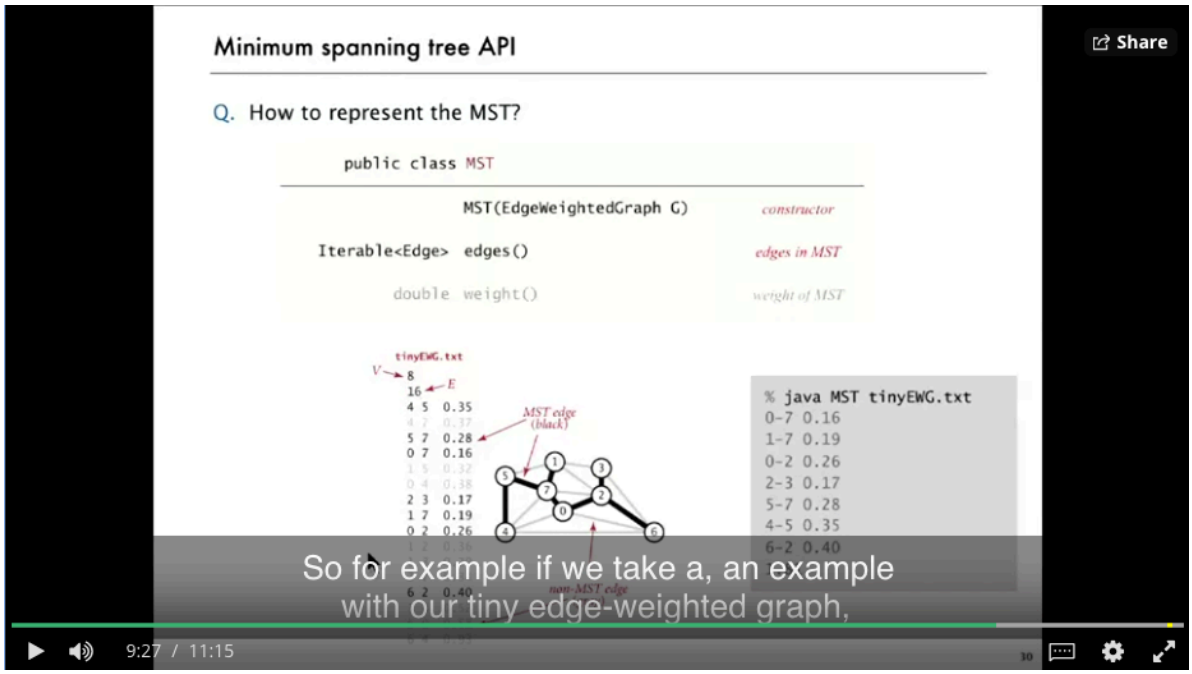
图的表示, 面向对象相关

有一个整数V, 然后维护一个长度为V的集合.



MST的表示

给一个图, 然后返回iteratorMST的节点或者权重



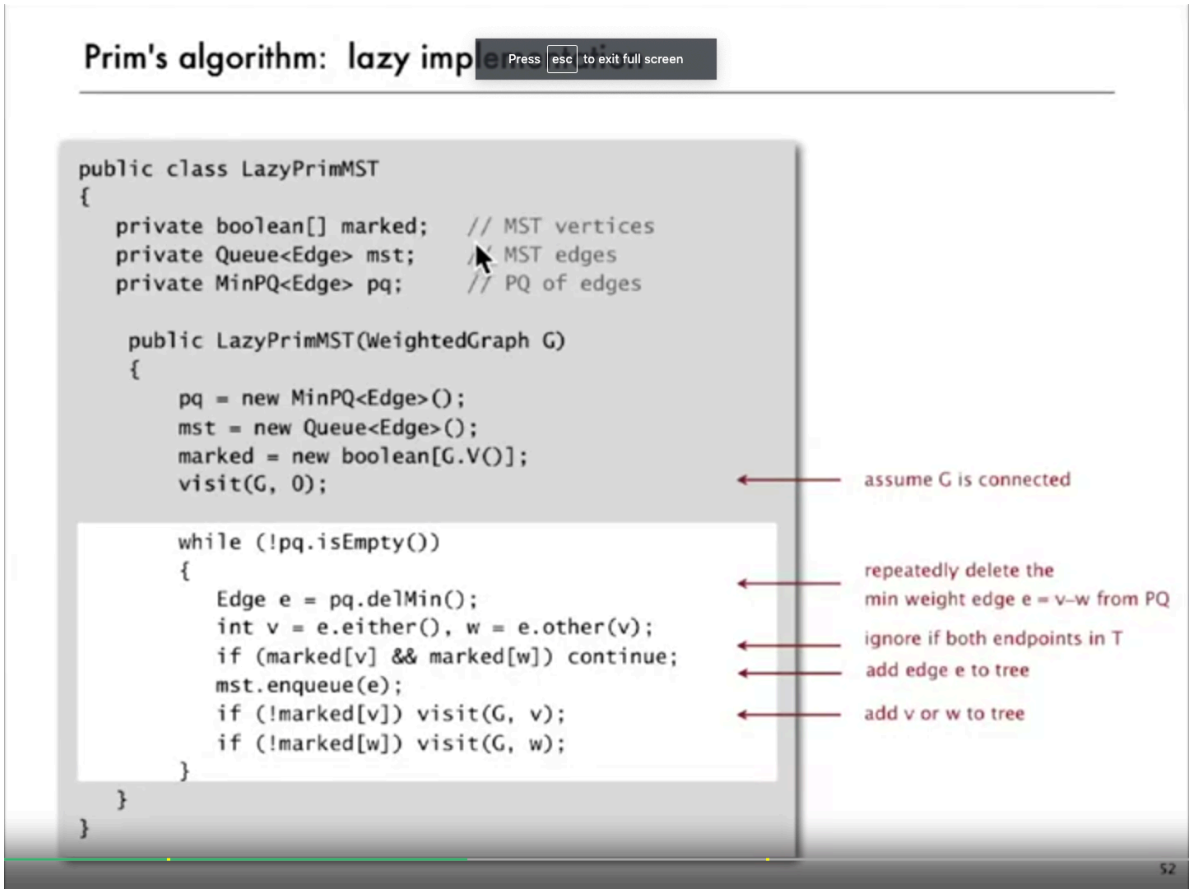
Kruskal's Algorithm(无像连通图)

首先sort所有的边(使用priorityQueue), 然后找到最短的, 看两个点是否标记, 没标记, 就把两个都标记. 入栈

Prim's Algorithm

MST用priority queue

minimum-cost spanning tree 最小生成树



先找一个节点标记mark, 然后用priorityQueue把这个节点所有的边enqueue进去, 然后根据priorityQueue拿出来最小的, 然后再把拿出来的节点标记mark, 把他的边enqueue. 如果poll的两个顶点都用过就continue. n - 1条边的时候停止.

union find

link解释的很好