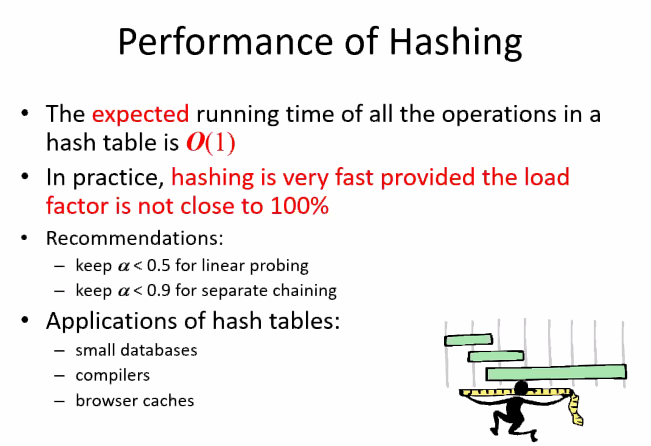
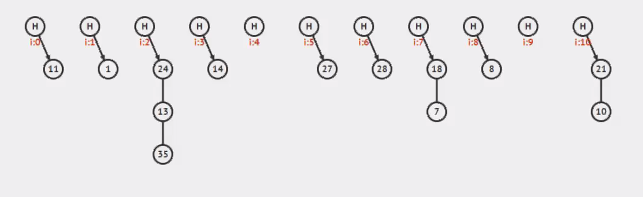
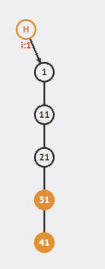
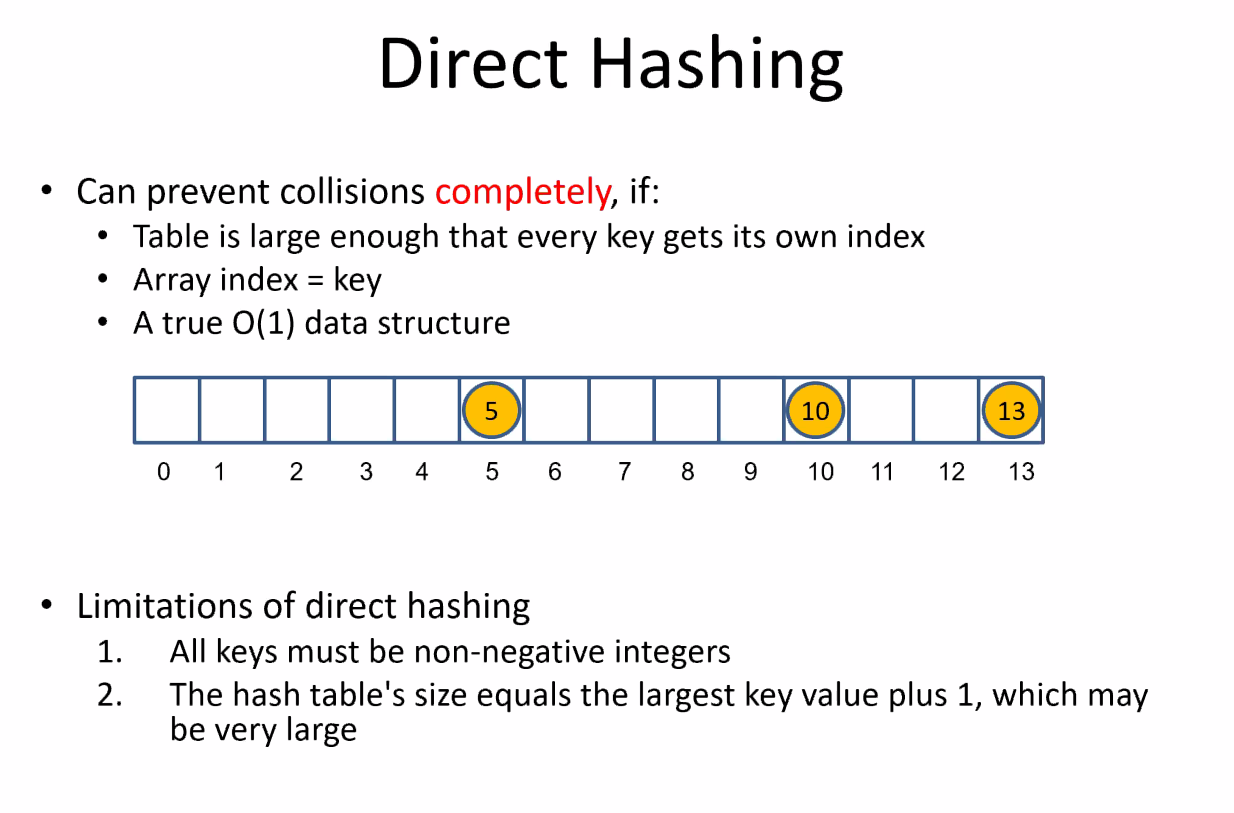
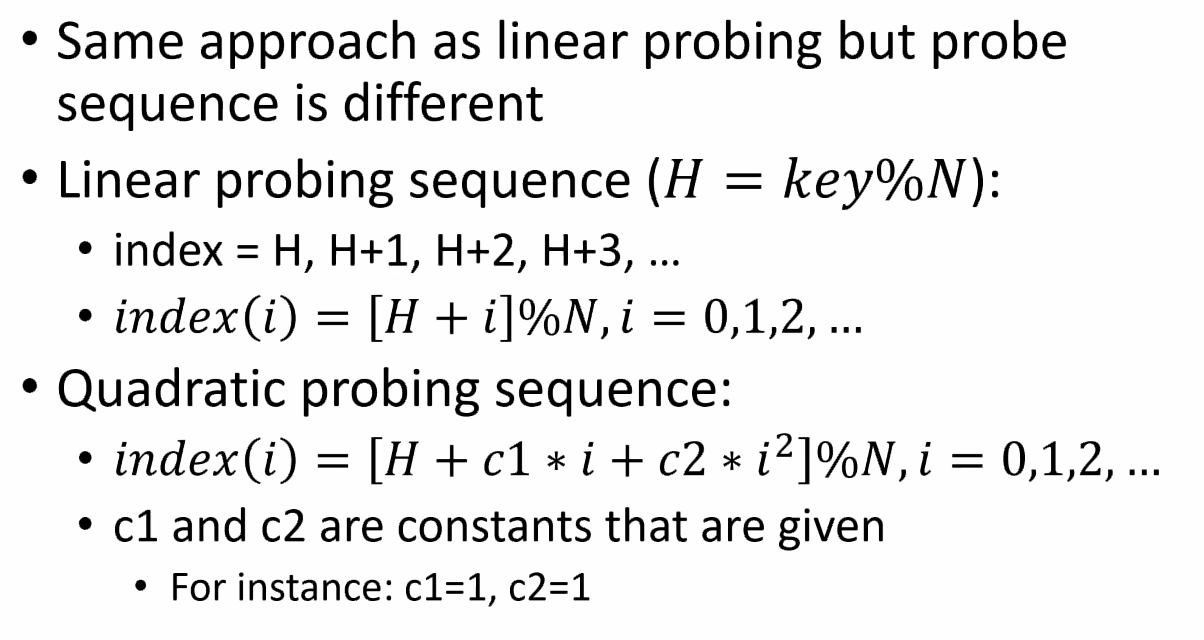
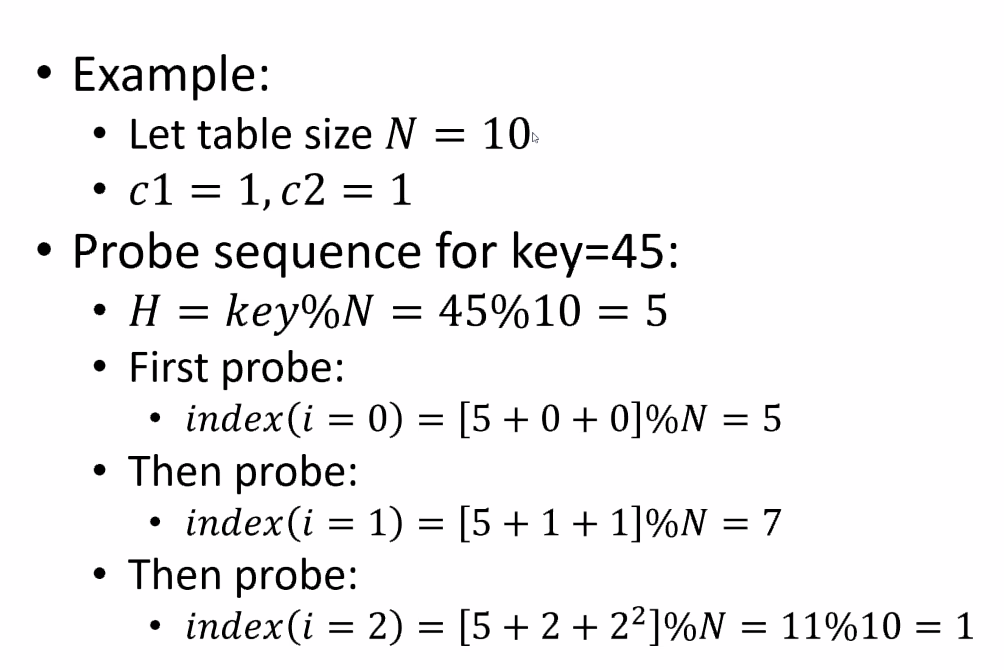
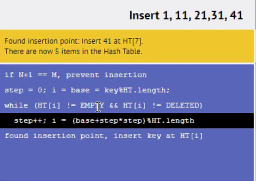
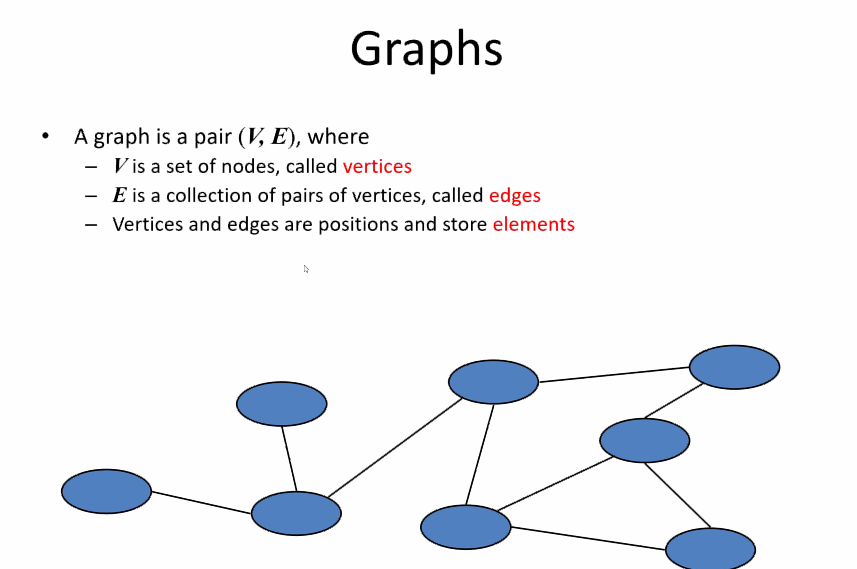
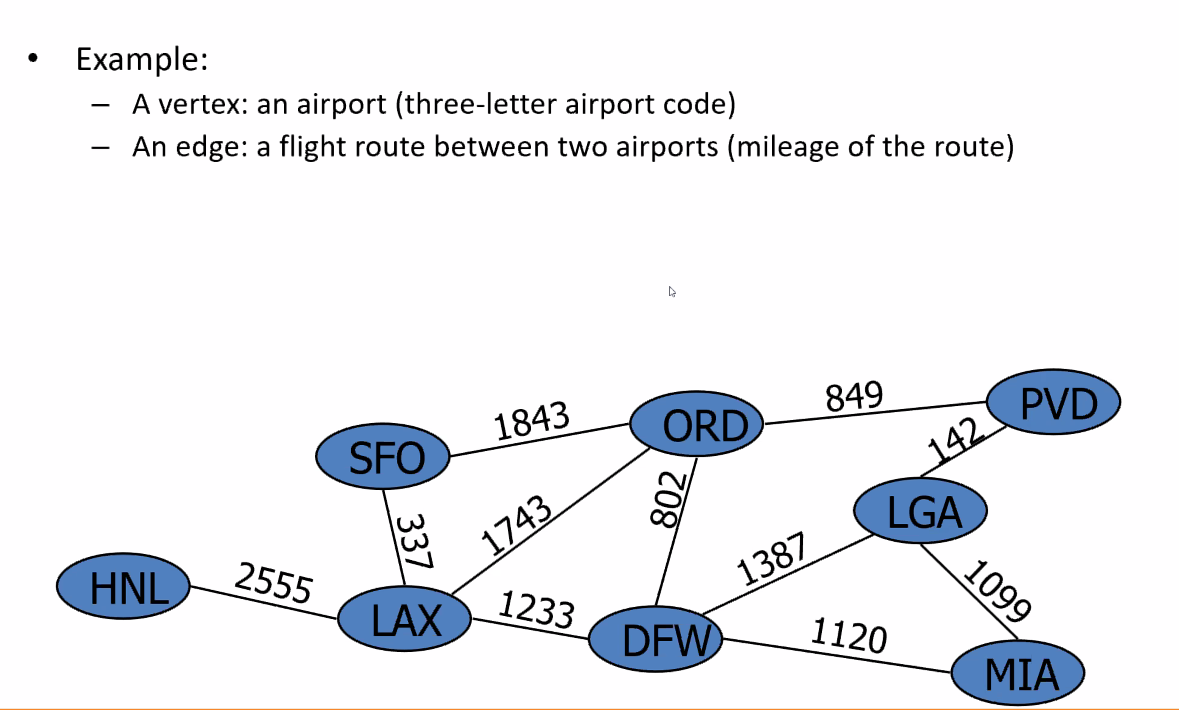
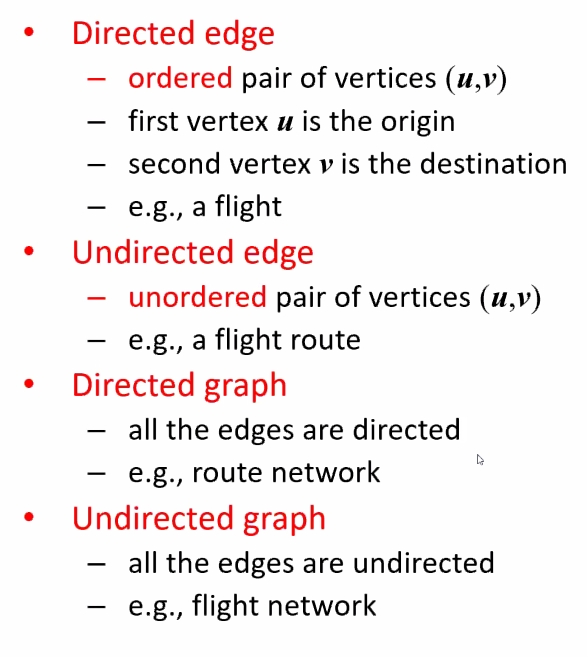
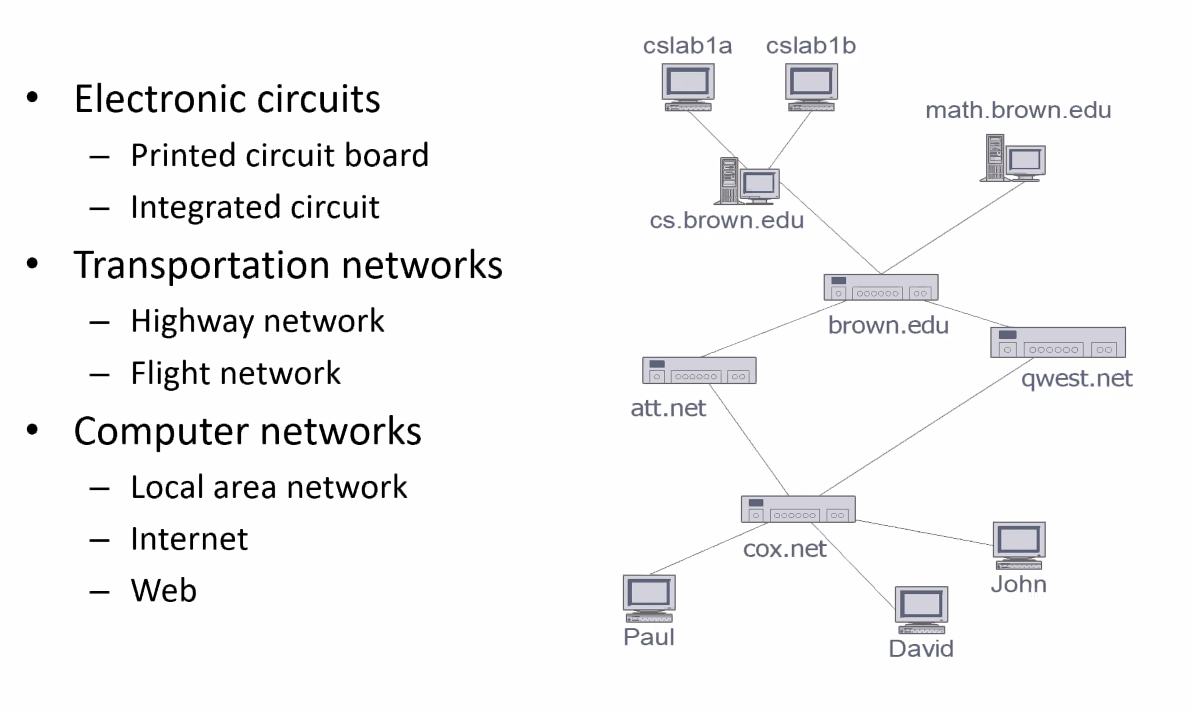
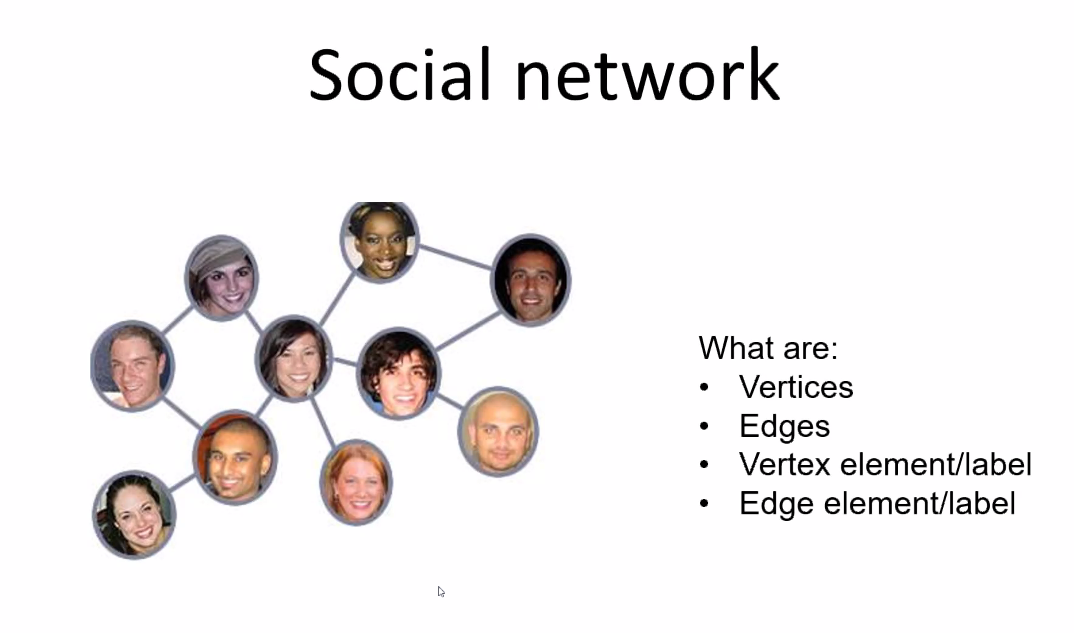
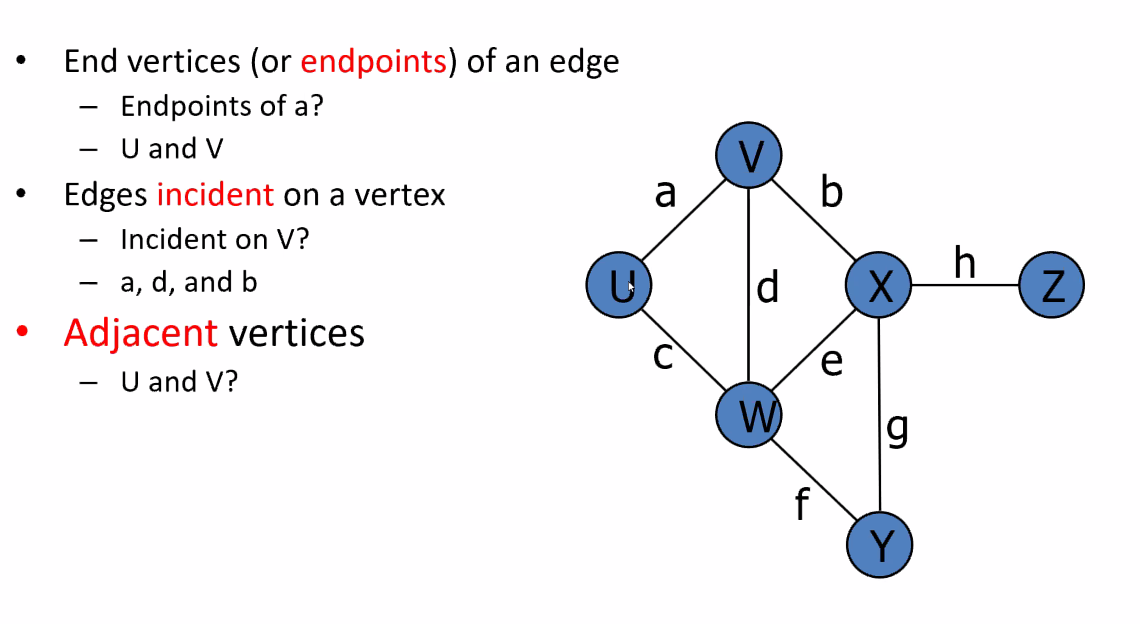
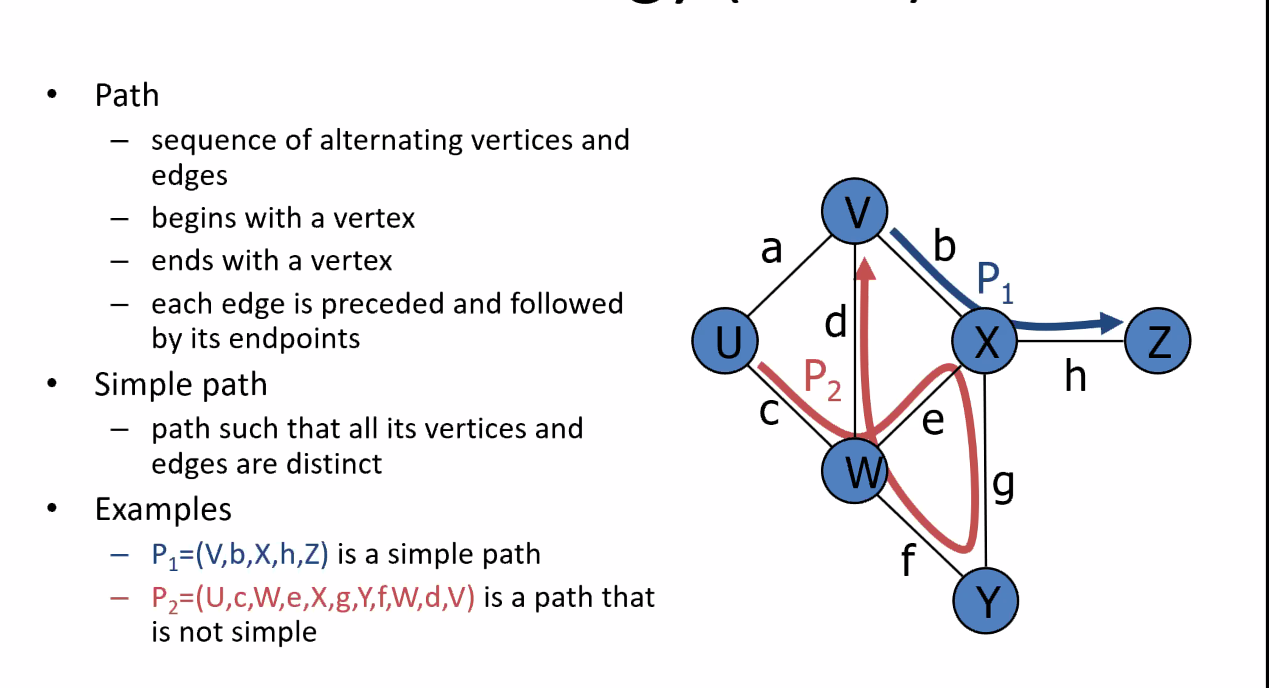
Lecture 25

CPSC 131

12/2/2020

1. Performance of Hashing
   1. 
      1. Go to visualgo.net/en for some pretty good visuals of data structures
   2. Chaining  
      
      1. When chaining reaches its limit  
           
         We rehash it by creating an empty table from 5 to 7 buckets
   3. Linear Probing
      1. Direct Hashing  
         
      2. The number of buckets does better if we use prime numbers
      3. It
      4. Is
      5. Important
      6. To know
      7. Priming prevents number of collisions from happening by dispersing out the buckets  
         
   4. Quadratic Probing
      1. 
      2. 
      3. Visualized as we use a formula to determine the next bucket   
           
         
      4. You might have to sketch this in a future quiz.
   5. Binary search tree vs hash table
      1. Example: You have a lot of students registering for the upcoming semester. What is the best way to organize them and then search for one?
         1. You want O(1) time for searching. Hash Table will always outperform a binary search tree.
         2. When it comes to insertion, you can expect a hash table to almost always get a worst-case of log(n). A Binary Search Tree guarantees that worst-case is log(n), leaving better-cases of O(n)/linear.
         3. Which structure is better is not the right question but rather what is best for the situation at hand.
2. Graphs
   1. 
   2. Example  
       
      1. Hey look, linear algebra!
   3. Edge types  
      
   4. Applications  
      
   5. Social Network  
      
   6. Terminology  
      Degree – 1+number of edges  
      
   7. Graph Representations  
      