CSCI 2270 Lecture Notes

3/20/19

Lecture cancelled Friday

Heapsort Big-O

1. building the heap
   1. O(log n) is cost for insert into almost full tree
   2. O(1) is cost of insert into empty tree
   3. n\*½log n = O(n\*log n)
2. remove all elements from heap
   1. O(n\*log n )
3. total complexity
4. n\*log n + n\*log n = O(n\*log n)

New Data structure: Graphs!

* Different from trees
  + all trees are graphs, but not all graphs are trees
  + less restricted
  + more complex
  + more degrees of freedom
  + no strictly defined parent/child relationship
* edges connect vertices
  + vertices are what we call nodes for graphs
  + connected vertices are adjacent

Social Media Platform

* called “FaceSpace” : network of interconnected users (sound familiar??)
* want to know exactly how many degrees of separation exist between any 2 users
* each user profile corresponds to a vertex
* a direct connection between two of the vertices is called an edge
* Traversal is done by moving between adjacent vertices
* Graph = {V, E}, were V is a set of vertices, and E is a set of edges, where the number of each do NOT have to be equal

Adjacency Matrix (unweighted type)

* 2D matrix
* all vertices appear as rows and columns
  + 1 in spot means they are adjacent
  + 0 means not adjacent

Weighted Type

* edges are given values, where a larger value represents a stronger connection with another vertex
* could be weighted in either direction, representing how strong the relationship is in both directions separately

Adjacency List

* each vertex gets a list associated with it, containing the vertices in which they are connected to
* can also contain weights